NASA Technical Memorandum 86230

Study of a Homotopy Continuation Method for Early Orbit Determination With the Tracking and Data Relay Satellite System (TDRSS)

R. L. Smith and C. Huang

MARCH 1986

NASA

NASA Technical Memorandum 86230

Study of a Homotopy Continuation Method for Early Orbit Determination With the Tracking and Data Relay Satellite System (TDRSS)

R. L. Smith and C. Huang
Computer Sciences Corporation
Beltsville, Maryland

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1986

PREFACE

The work described in this document was performed under Contract NAS 5-27600, Task Assignment 80900.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the contributions of J. Teles, C. Doll, Jr., and D. Cowan of Goddard Space Flight Center (GSFC) to this study.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Figure

LIST OF ILLUSTRATIONS (Cont'd)

Figure

SECTION 1 - INTRODUCTION

The purpose of early orbit determination methods is to derive a set of orbit elements that match available observations, when, initially, the orbit elements are not well known or not known at all. Characteristically, early orbit determination methods use approximate physical models and observations collected over a limited time span, usually less than one orbital period, in order to accelerate processing. Early orbit methods are a necessary part of orbit operations procedures. As NASA converts its spacecraft tracking from the ground-based system (Ground Spaceflight Tracking and Data Network (GSTDN)) to a satellite relay system (Tracking and Data Relay Satellite System (TDRSS)), it is necessary to have a reliable early orbit method available in the GSFC Flight Dynamics Facility (FDF) that functions with TDRSS tracking. This memorandum reports on the development and verification of such a method.

Currently existing early orbit methods make use of angular antenna pointing observations collected at the ground stations. However, as is discussed in Section 2, the open loop TDRSS angular antenna pointing observations are too inaccurate for use in even early orbit determination; therefore, an early orbit method that uses the precise TDRSS range and Doppler tracking exclusively is required. Since the problem is basically one of solving a set of nonlinear equations, which specify that the predicted observations match the actual ones, the significant mathematical advances of the last 10 years in this area seemed to be applicable to the problem. It appeared that one of these recent advances, the homotopy continuation method for solving nonlinear systems of equations, was particularly well suited for early orbit determination, where a priori estimates of the solution are often quite inaccurate or not available. This memorandum

centers on the formulation and testing of the homotopy continuation method for orbit determination using TDRSS.

Section 2 presents a description of the TDRSS early orbit problem and indicates characteristics that are desirable in an early orbit algorithm. Section 3 gives an extended discussion, with simple examples for illustration, of the particular formulation of the homotopy continuation method that was studied. This section also includes a description of the trajectory and observation models used in this study, as well as a detailed description of the numerical computational algorithm that was developed. Section 4 presents two detailed numerical examples, one with real TDRSS tracking and the other with simulated tracking. Section 5 considers a particular implementation of the early orbit algorithm, one that is fairly complete and automatic, and measures its performance over a large number of Monte Carlo trials.

The main drawback of the formulation of the homotopy continuation method that is given in Section 2 is the occurrence of disjoint solution loops. Section 6 develops a generalization that remedies this problem; Section 6 also indicates a remaining problem to be solved if the method is to be applied in least-squares orbit determination. Sections 2 through 5 consider only the six-observation case.

The conclusion, Section 7, provides a brief summary of results found in this study, and suggests main directions for additional enhancements of the method.

SECTION 2 - THE TDRSS EARLY ORBIT DETERMINATION PROBLEM

This section describes the general nature of the TDRSS early orbit determination problem and the characteristics that are desirable in a solution method. The main difference between early orbit determination with TDRSS and with many other tracking systems is that the open loop angle information available is relatively crude in TDRSS. Also, small angular errors from geosynchronous distances map into large position errors for low-altitude spacecraft. Consequently, standard existing algorithms (Reference 1), which rely heavily on angle measurements, are not directly applicable, and additional methods must be devised for early orbit determination for the case of pure range and Doppler tracking.

The TDRSS relay range and Doppler measurements have a high precision; analyses performed at GSFC indicate typical measurement noise standard deviations of about 0.5 meters and 0.5 millimeters per second, respectively. However, the reported values of the Tracking and Data Relay Satellite (TDRS) antenna beam angles are not actual measurements (in the S-band Single Access (SSA) mode and the S-band Multiple Access (MA) mode), but, rather, they are the predicted angles for open-loop antenna pointing and are based upon the predicted trajectory of the target spacecraft. Range and Doppler tracking can be performed only when the target spacecraft lies within the TDRS antenna beam. In the SSA mode, the antenna full beam-width is 1.9 degrees, while for the MA mode, it is about 3.0 degrees. For the K-band Single Access (KSA) mode, in which closed-loop antenna pointing is used, the antenna full beam-width is much smaller, 0.44 degrees. (However, the KSA mode is not used for applications where large errors are likely because of the small acquisition angle errors, which must be adhered to. Also, most TDRSS user spacecraft do not have K-Band capability.) For

comparison, the full-width of the Earth, as seen from a TDRS, is about 17.4 degrees. (See Figure 2-1).

Typical circumstances under which an early orbit determination method may be required are as follows.

Consider a spacecraft which, after some period of routine orbit maintenance, is scheduled for an orbit maneuver or, consider a malfunction of an expendable launch vehicle. Hypothetically, because of a malfunction in the thrusters themselves or in attitude control, the actual vector thrust might turn out to be different than was planned. Since open-loop antenna pointing is based upon the planned orbit rather than the actual orbit, the spacecraft might stay within the antenna beam pattern, and relay range and Doppler measurements may be collected for some short periods of time; the length of time would depend upon the severity of the malfunction before contact is lost. Perhaps, through continued attempts to find the spacecraft, isolated additional contacts might be made, and some additional tracking may be collected. At this point, with or without the additional contacts, an early orbit determination method would be required to determine the orbit from the limited available data and poor a priori knowledge of the orbit elements.

The orbit determination error that is caused by the uncertainty of the spacecraft position within the antenna beam can be large. This is illustrated in Tables 2-1, 2-2, and 2-3. In these tables, orbits that correspond to locations near the edges of the SSA antenna beam are compared with orbits that correspond to the beam center. These tables show the error that can result if the errant spacecraft is assumed to be at the beam center and an orbit is then derived using this (incorrect) assumption. Tables 2-1 and 2-2 are for low-altitude, circular orbits at low and high inclinations, while Table 2-3 is for a low-inclination, highly

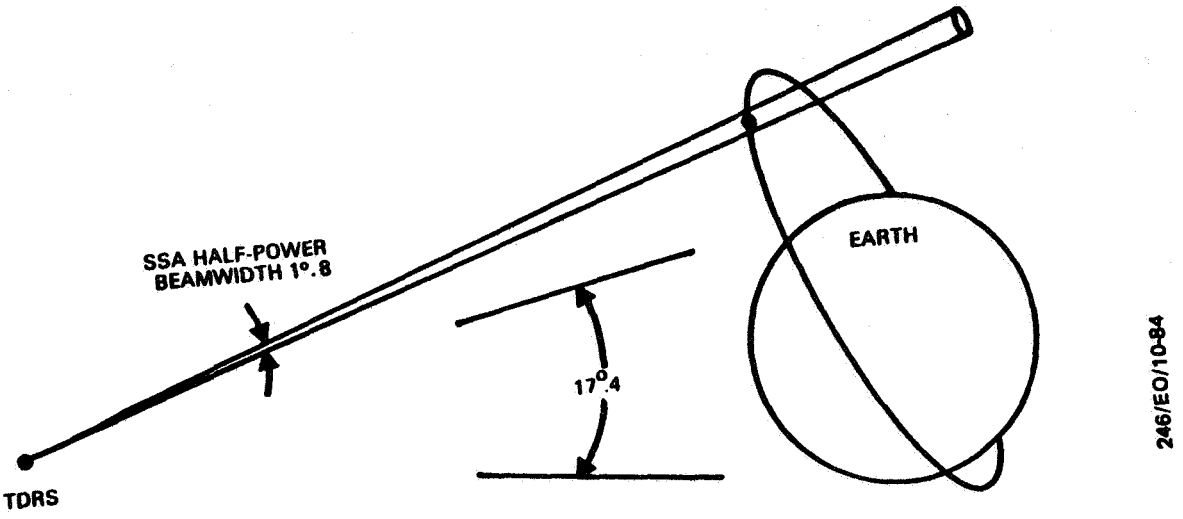


Figure 2-1. Diagram Showing the Relative Angular Sizes of the Earth and the TDRS SSA Antenna Beam

Table 2-1. Orbit Determination Error Induced by TDRS Antenna Beam Angle Uncertainty (Low-Inclination, Circular Orbit)

Kepler Element	Orbit Elements For Center of TDRS Antenna Beam	Orbit Elements for Positions Offset From Center of TDRS Antenna Beam[1]			
		ΔAZ[2] = $0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = -0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = 0.^{\circ}86$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = -0.^{\circ}86$
a(km)	7000	6715	7335	6942	7119
e	0.001	0.076	0.074	0.011	0.0087
i(deg)	10	10.0	10.0	6.2	15.6
Ω(deg)	145	144.4	145.6	186.6	130.2
ω(deg)	0	135.6	-46.3	297.9	183.2
M(deg)	190	48.6	242.3	210.7	220.9
Maximum One-Revolution Error (km)		2912	2911	979	1279

[1]The offsets from the beam center are the same at both times that are used to define the orbit (t = 0 and t = 25 minutes). The range measurements for the offset orbit are assumed to be the same as for the beam-centered orbit.

[2]Azimuth (AZ) and Elevation (EL) are measured at the TDRS. Azimuth is defined to be the projected angle in the TDRS orbit plane, zero at the nadir, and positive toward the east. Elevation is positive at positions north of the TDRS orbit plane.

Table 2-2. Orbit Determination Error Induced by TDRS Antenna Beam Angle Uncertainty (High-Inclination, Circular Orbit)

Kepler Element	Orbit Elements For Center of TDRS Antenna Beam	Orbit Elements for Positions Offset From Center of TDRS Antenna Beam[1]			
		ΔAZ[2] = $0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = -0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = 0.^{\circ}86$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = -0.^{\circ}86$
a(km)	7000	7051	7012	6613	7455
e	0.001	0.012	0.012	0.053	0.053
i(deg)	100	105.7	93.9	101.4	98.9
Ω(deg)	55	58.6	51.4	55.7	54.4
ω(deg)	0	67.7	-92.9	204.1	18.2
M(deg)	260	194.4	351.9	49.1	248.8
Maximum One-Revolution Error (km)		900	925	3587	3634

[1]The offsets from the beam center are the same at both times that are used to define the orbit (t = 0 and t = 25 minutes). The range measurements for the offset orbit are assumed to be the same as for the beam-centered orbit.

[2]Azimuth (AZ) and Elevation (EL) are measured at the TDRS. Azimuth is defined to be the projected angle in the TDRS orbit plane, zero at the nadir, and positive toward the east. Elevation is positive at positions north of the TDRS orbit plane.

Table 2-3. Orbit Determination Error Induced by TDRS Antenna Beam Angle Uncertainty (Low-Inclination, High-Eccentricity Orbit)

Kepler Element	Orbit Elements For Center of TDRS Antenna Beam	Orbit Elements for Positions Offset From Center of TDRS Antenna Beam[1]			
		ΔAZ[2] = $0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = 0.^{\circ}86$, $\Delta EL = 0.^{\circ}0$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = 0.^{\circ}86$	$\Delta AZ = 0.^{\circ}0$, $\Delta EL = -0.^{\circ}86$
a(km)	12500	11848	13194	12432	12590
e	0.44	0.45	0.43	0.45	0.43
i(deg)	10	10.1	9.9	9.0	11.2
Ω(deg)	145	144.4	145.6	155.1	136.9
ω(deg)	0	4.1	-4.3	-10.3	8.3
M(deg)	225	218.4	231.6	225.4	224.7
Maximum One-Revolution Error (km)		2505	2365	808	883

[1]The offsets from the beam center are the same at both times that are used to define the orbit (t = 0 and t = 25 minutes). The range measurements for the offset orbit are assumed to be the same as for the beam-centered orbit.

[2]Azimuth (AZ) and Elevation (EL) are measured at the TDRS. Azimuth is defined to be the projected angle in the TDRS orbit plane, zero at the nadir, and positive toward the east. Elevation is positive at positions north of the TDRS orbit plane.

eccentric orbit. In each table, four different angular offsets near the edge of the SSA antenna beam are considered, the same offset being applied at t = 0 and t = 25 minutes. The range measurements at these two times are assumed to be the same for the beam-centered and offset orbits.

(In Tables 2-1, 2-2, and 2-3, an orbit is considered to be determined by two sets of azimuth, elevation, and range.) Errors will be correspondingly larger in the MA tracking mode because of the larger antenna beamwidth.

As indicated by the tables, the SSA one-revolution position errors can easily be as large as 1000 or 2000 kilometers. The uncertainty in angular position causes errors of this size for three main reasons: (1) error in the orbit plane orientation, (2) error in the true anomaly difference between the two derived radius vectors, and (3) error in the magnitudes of the two derived radius vectors. The latter two effects are illustrated in combination by Figure 2-2. Effect (1) can cause very large errors when the true anomaly difference is close to an integral multiple of 180 degrees, while effects (2) and (3) will become especially sensitive to angular errors for small values of the true anomaly difference.

From the results in the tables, it is clear that the TDRS antenna beam angles cannot be used as primary data in determining the orbit. However, this data is useful in defining an a priori estimate of the orbit, which can then be used to initialize calculation of a final solution that is based on precise range and Doppler data alone.

Although the range and Doppler measurements are sufficiently precise, there is a mathematical problem in performing orbit determination exclusively with such measurements. The problem is that six range and Doppler measurements do not uniquely determine the orbit; that is, the problem can have

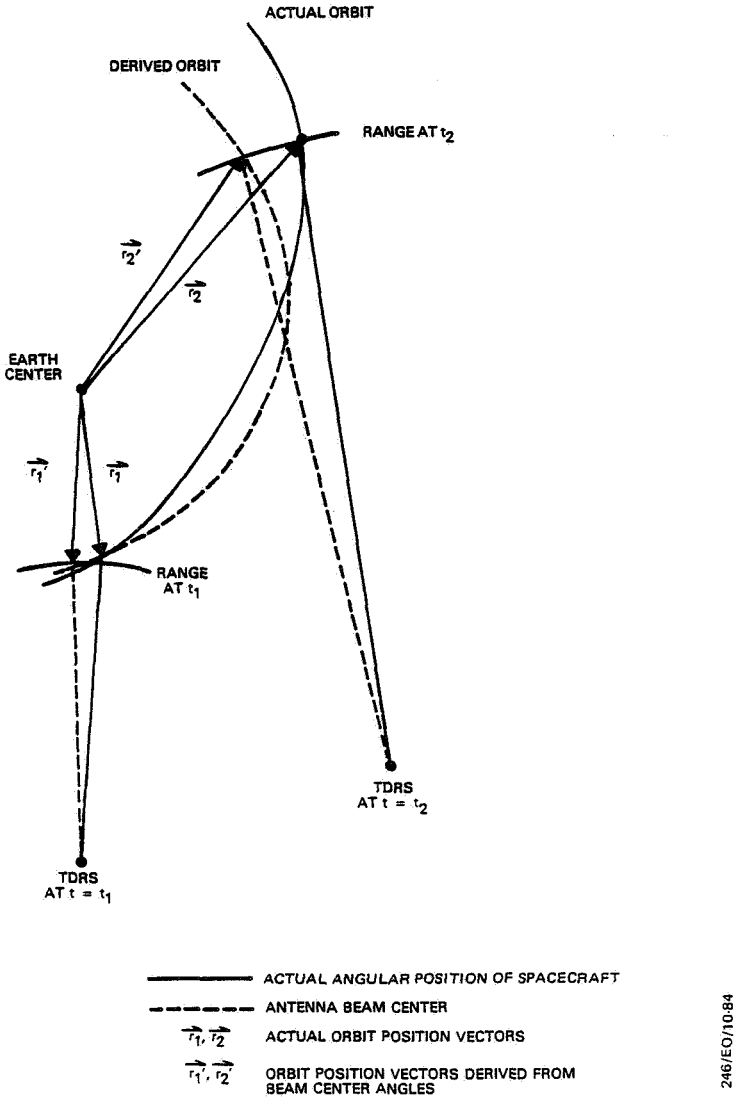


Figure 2-2. Schematic Diagram of the Effect of Angular Position Errors on the Determined Orbit

several distinct solutions. In the TDRSS tracking configuration, four distinct orbit solutions are typically found, and, in this study, up to 12 solutions have been encountered in a small number of cases. This multiplicity would persist even in a least-squares formulation. Solution multiplicity leads to reduced radii of convergence near the "correct" solution when iterative methods, such as the Newton-Raphson method, are used to solve the equations numerically. Because of solution multiplicity, the TDRSS early orbit method must contain a technique for systematically isolating and collecting the solutions for the given set of observations and for testing each of them against additional constraints, such as the antenna beam angles or additional range and Doppler data, as well as reasonableness of the orbital elements. For example, large plane changes and large energy changes may be beyond the ΔV capability of the vehicle, and some solutions can be easily discarded.

In addition to handling the multiple solution problem, the early orbit method should possess the following general and specific characteristics:

1. The method should always yield the correct orbit solution with valid tracking data, regardless of the error in the a priori orbit estimate; that is, it should construct the correct solution given any a priori estimate. (The term "correct" solution requires some clarification. With sufficient range and/or Doppler tracking, only a finite number of orbit solutions exists, except for special geometrical configurations. One solution in this finite set corresponds to the actual orbit in the sense that as the observation and modeling errors are continuously reduced to zero, this "correct" solution is the one that approaches the actual orbit.)

2. The method should determine the orbit without numerical problems, given any mathematically sufficient

distribution of tracking. Special geometrical tracking configurations and tracking time distributions must not be required. That is, if an observation set can define the orbit, the method, or, at least a finite number of possible orbits, then the method should be capable of finding the solution.

3. It would be desirable for the method to utilize any of the precise TDRSS tracking types, including range only, Doppler only, mixed range/Doppler differenced Doppler, and hybrid range and Doppler types. Data from any of the TDRS's, or in combination, should be usable.

4. The availability of a good a priori orbit estimate should speed the determination of the solution.

5. The method should have a capability to refine its initial two-body solution using improved physical models for the trajectory and for the measurements. This will enable subsequent TDRSS acquisition of the spacecraft by providing a good predicted trajectory.

6. The algorithm must redetermine the range ambiguity numbers in the case of a very poor a priori orbit estimate. For S-band tracking, the range ambiguity distance is approximately 13,000 kilometers, which means that points along the line of sight at intervals of 13,000 kilometers from the initial, ambiguous distance, are candidates for the unambiguous value of the range. (This redetermination should be necessary only rarely, because an orbit so far in error is unlikely to fall within the limits of the predicted TDRS antenna beam angles.)

7. For the Space Shuttle and other applications, the algorithm must succeed without range tracking; only Doppler tracking and the antenna beam angles will be available.

A three-stage early orbit determination method is considered in this report. The first stage, using range measurements and using the antenna beam angles as measurements, derives a preliminary solution from any a priori orbit estimate. The second stage, using only range and/or Doppler measurements and the first-stage solution as an a priori estimate, determines an intermediate solution, which is based, like the first stage, on two-body dynamics and simple measurement modeling. The third stage uses the intermediate solution as the a priori estimate and uses the same range and/or Doppler measurements, but improves the physical accuracy by using more precise trajectory and measurement modeling to derive the final solution. If a good a priori orbit estimate is available, the first stage can be bypassed. If high accuracy in the solution is not needed, then the third stage can also be bypassed.

The primary computational algorithm in each of these three stages is the algorithm for solving a system of nonlinear algebraic equations. The homotopy continuation method was selected to perform this function. This general method can be applied to a large variety of problems; its application to orbit determination is described in the next two sections.

SECTION 3 - FORMULATION OF THE BASIC HOMOTOPY METHOD

In Section 3.1, the formulation of the homotopy method, as applied to spacecraft orbit determination, is described in general terms. Section 3.2 contains the details of the observation models and partial derivatives that were used in the developmental computer program. In Section 3.3, a very simple example that requires a solution space of only one dimension, rather than six, is presented. Section 3.4 explains the relationship between the standard Newton-Raphson method for solving systems of equations and the homotopy method. Finally, Section 3.5 describes the numerical algorithm that was developed for following the solution curves.

3.1 GENERAL FORMULATION

Six observations O_i^1, $i = 1,\ldots 6$, are selected to determine the orbit. These six observations may be of any type and in any combination. Although no special time distribution is required, if observations of the same type are too close together, the determined orbit may have large errors (or may not even exist) because of measurement errors, whereas, if the observations are spread over too large a period of time, the error introduced by the computationally efficient modeling simplifications may become too large. The superscript "1" on the symbol for the observations will indicate that they are the given, real observations.

Next, an estimate of the solution state vector, $\vec{x}^0$, at the reference time is selected. It is assumed that an observation and dynamics model represented by the functions C_i, $i = 1,\ldots, 6$, is available to relate any reference time state vector, x, to the modeled observations, which correspond in type and time to the real observations O_i^1.

Denoting by O_i^0 the modeled observations corresponding to estimate $\vec{x}^0$, the estimate satisfies the following equations, by construction:

$$O_i^0 - C_i(\vec{x}^0) = 0, \quad i = 1, \ldots, 6 \qquad (3\text{-}1)$$

On the other hand, an unknown solution state $\vec{x}^1$ satisfies the equations

$$O_i^1 - C_i(\vec{x}^1) = 0, \quad i = 1, \ldots, 6 \qquad (3\text{-}2)$$

where the functions C_i are the same as in Equations (3-1), since the observation times are assumed to be fixed in this formulation.

The homotopy continuation method smoothly deforms the left-hand sides of Equations (3-1) into the lefthand sides of Equations (3-2), using the parameter λ and permitting solution states, x, for all applicable values of λ. This smoóth deformation is described by the equations

$$[1 - \lambda]\left[O_i^0 - C_i(\vec{x})\right] + [\lambda]\left[O_i^1 - C_i(\vec{x})\right] = 0, \quad i = 1, \ldots, 6$$

or equivalently

$$\left[O_i^0 + \lambda\left(O_i^1 - O_i^0\right)\right] - C_i(\vec{x}) = 0, \quad i = 1, \ldots, 6 \qquad (3\text{-}3)$$

These equations define the homotopy, that is, a continuous mapping from one mathematical function to another. At $\lambda = 0$, Equations (3-3) have the known solution x^0, while at $\lambda = 1$, the equations have the solution sought. In the seven-dimensional λ - x space, the set of solutions of

Equations (3-3) form a smooth curve that passes through the estimate, $\vec{x}^0$, and the desired solution, $\vec{x}^1$.

It should be remarked that the formulation described here is quite general, and in principle any modeling, including high-precision modeling, can be used for the observations and for the dynamics. For most applications, however, efficiency suggests the use of approximate models, reserving high-precision levels until after a satisfactory interim solution has been obtained.

To solve Equations (3-3), a numerical algorithm (see Section 3.4) is used to follow the solution curve from the estimate through the solutions.

Although this report considers TDRSS tracking exclusively, there is no essential difference if, instead, ground-based tracking is employed. The only requirement is that the position and velocity of the tracker, whether it is a ground station or a relay satellite, be known at the times of the tracking measurements. In principle, the formulation given in this section is readily extended to include additional parameters in the solution state, for example, TDRS orbital elements, ground station coordinates, measurement biases, and spacecraft force model parameters. However, this generalization is left to future study.

The theoretical basis for the homotopy method expressed by Equations (3-3) is developed in Reference 1. Reference 2 begins with a very brief statement of this theoretical basis and continues with many examples from engineering problems. (However, orbit determination is not included among the examples.) Reference 3 is a review article, which includes the homotopy continuation method in addition to simplicial methods (triangulation networks) for finding roots. A mathematically correct and clear discussion of the homotopy method in orbit determination would require the language and

results of differential topology and will not be undertaken here. Rather, the approach taken is a qualitative description of the solution curves of Equations (3-3), based primarily on the results of many numerical experiments.

A solution curve for Equations (3-3) in the case of pure TDRSS relay range and/or Doppler tracking is schematically illustrated by Figure 3-1. In general, the solution curve for a given orbit determination problem (that is, the specification of the numerical values of six range and/or Doppler observations and the numerical value of the a priori estimate) consists of a number of disjoint smooth, closed curves, or loops. When the a priori estimate is appropriately related to the desired solution, then both states will lie on the same loop. The set of orbit solutions for the given problem consists of the collection of all the intersection points of the loops with the $\lambda = 1$ hyperplane. If the a priori estimate is changed, keeping the six observations fixed, then the number and shape of the loops may change, but the number and numerical values of the solution states do not change.

For very special, isolated values of the a priori estimate, two of the loops may just touch each other. This is illustrated by Figure 3-2, which schematically shows how a solution curve may split into two curves as the a priori state is allowed to change and to pass through such a special value. These special values are unlikely to occur in practice, and this unlikelihood corresponds to the statement that the solution curves are smooth (and therefore do not have such touching points) "with probability one," a result that is proven in Reference 2.

As indicated by Figures 3-1 and 3-2, orbit states do not necessarily exist for every intermediate value of λ in range/Doppler orbit determination. When they do not

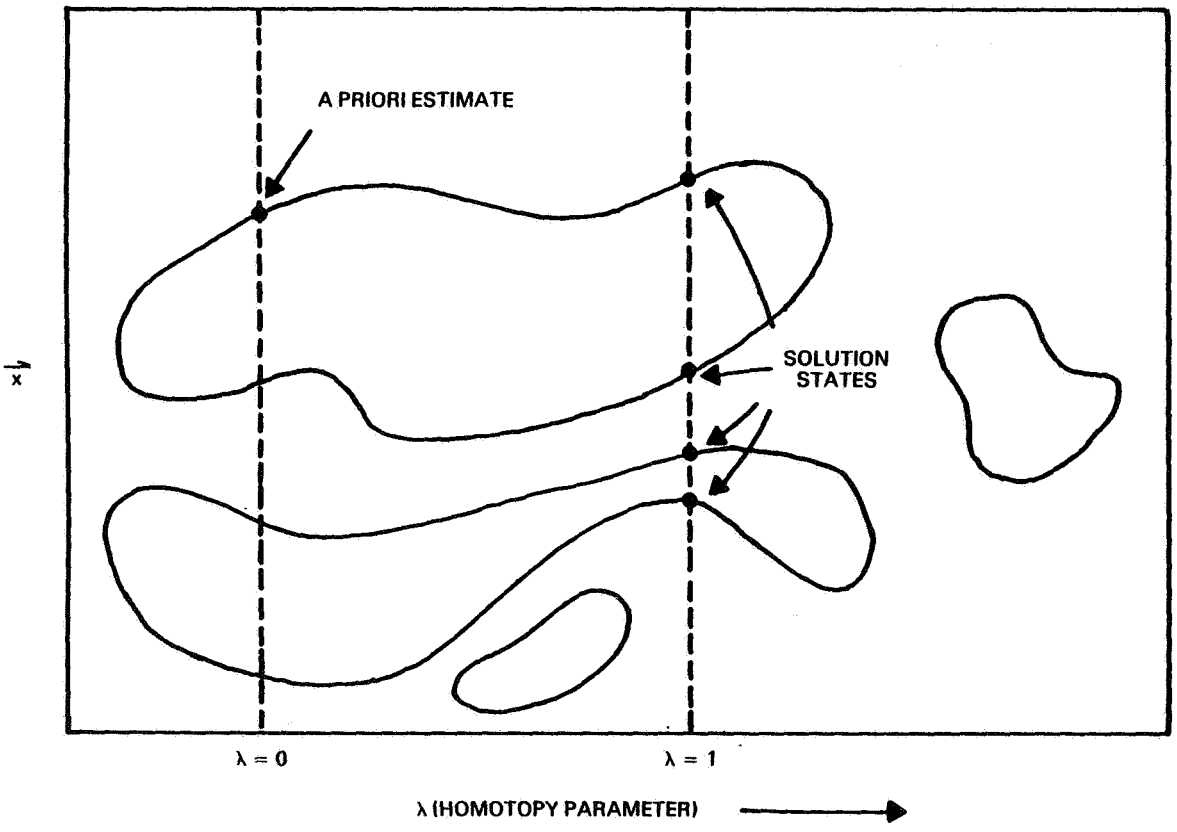


Figure 3-1. Schematic Diagram of Solution Curve in λ-$\vec{x}$ Space for TDRSS Range and/or Doppler Tracking

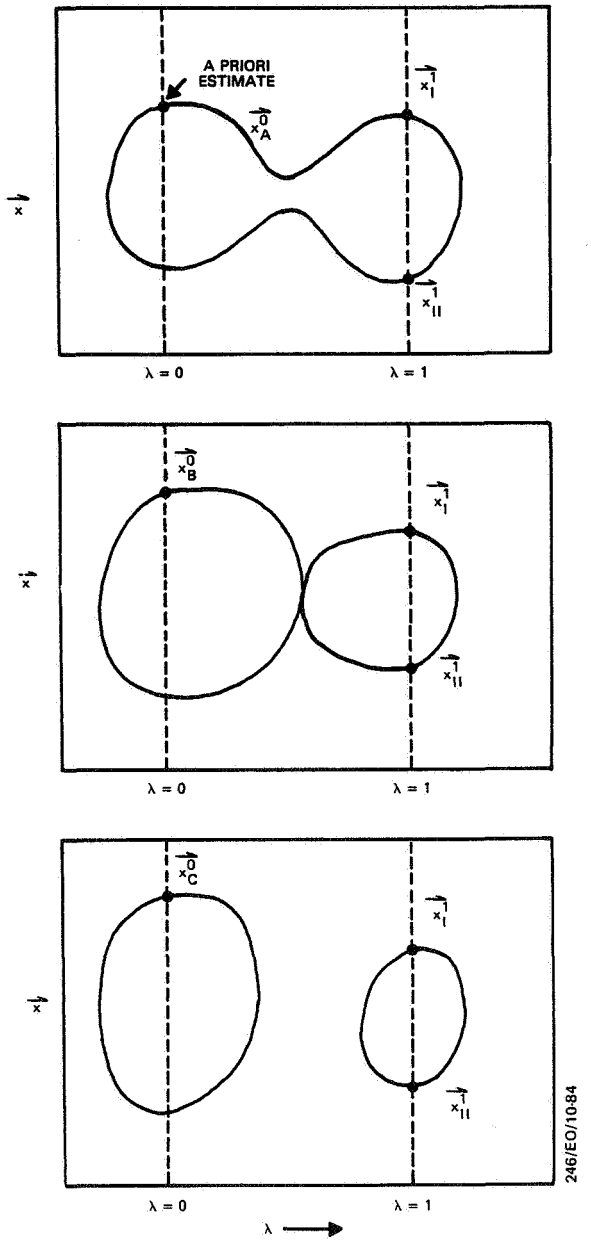


Figure 3-2. Schematic Diagram Showing the Resulting Sequence of Solution Curves for a Special Sequence of A Priori Estimates

disjoint loops will occur. This is a result of the physical fact that a solution state does not necessarily exist for six arbitrary numerical values for the range and Doppler measurements, which lie between the O_i^0's and the O_i^1's.

Finally, the solution curves are considered for the orbit determination problem in which, effectively, the three components of the spacecraft position vector are measured at each of two distinct times. In TDRSS, crude knowledge of the two position vectors comes from knowledge of the range and the antenna beam angles at two measurement times. Two position vectors are also equivalent to three simultaneous range measurements taken at three different trackers (trilateration) at each of two times, and this situation can be thought of as a limiting case in range-only orbit determination. This limiting case, however, is distinct from the general range-only case in that orbit solutions exist for every value of λ; that is, given any two position vectors, there are always (two-body) solution states that fit them. Thus, as indicated by Figure 3-3, the solution loop of the range-only orbit determination problem has expanded indefinitely, and the branches of the solution curve cover, without gaps, all values of λ from $-\infty$ to $+\infty$.

3.2 DETAILED MATHEMATICAL FORMULATION

In this section the mathematical formulas defining the TDRSS tracking measurement models used in this study are specified. The formulas for the partial derivatives are also specified. Reference 5 gives details of the TDRSS range and Doppler tracking system. Details of the TDRSS algorithms for data reduction and observation models are given in Reference 6.

In the developmental program, a reference time is selected for each case, and the variable t then denotes the time

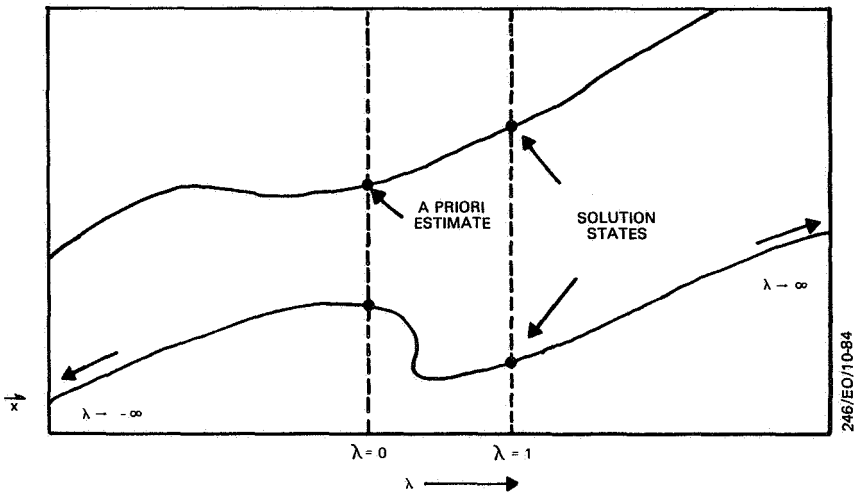


Figure 3-3. Schematic Diagram Showing the Solution Curve for Orbit Determination Based on the Measurement of Two Position Vectors on the Orbit

elapsed from the reference time. Six-component, cartesian orbit states, x, are always assumed to be referenced to t = 0. However, the measurement models require the six components of the orbit state, $\vec{y}$, at the measurement time to compute the numerical value of the measurement, $C(\vec{y})$. The relation of $\vec{y}$ to $\vec{x}$ is obtained through the orbit propagator (Section 3.2.4) and, correspondingly, measurement partial derivatives are computed using

$$\frac{\partial C}{\partial x_i} = \sum_{k=1}^{6} \frac{\partial C}{\partial y_k} \frac{\partial y_k}{\partial x_i}, \quad i = 1, \ldots, 6 \tag{3-4}$$

3.2.1 TDRSS RELAY RANGE MEASUREMENT

3.2.1.1 Geometric Range and Partial Derivative

The geometric model for the single-relay range measurement is the following:

$$\rho = |\vec{r} - \vec{R}_{TDRS}| + |\vec{R}_{TDRS} - \vec{R}_{WS}| \tag{3-5}$$

where ρ = geometric range measurement

$\vec{r}$ = position vector of the target spacecraft at time t (see Section 3.2.4)

$\vec{R}_{TDRS}$ = position vector of the TDRS at time t (see Section 3.2.4)

$\vec{R}_{WS}$ = position vector of the White Sands ground station at time t (see Section 3.2.5)

Assuming that $\vec{R}_{WS}$ and $\vec{R}_{TDRS}$ remain fixed throughout the calculations for a given observation, the local partial derivatives needed in Equation (3-4) are given by

$$\frac{\partial \rho}{\partial y_i} = \frac{r_i - R_{TDRS,i}}{|\vec{r} - R_{TDRS}|}, \quad i = 1, 2, 3$$

$$\frac{\partial \rho}{\partial y_i} = 0, \quad i = 4, 5, 6 \tag{3-6}$$

3.2.1.2 Modeling of the Light-Time-Correct Range

The light-time correct range, for any one of the four legs, satisfies the two equations

$$t_1 = t_2 - \frac{1}{c}\rho_{12} \tag{3-7}$$

$$\rho_{12} = \left|\vec{r}_2(t_2) - \vec{r}_1(t_1)\right| \tag{3-8}$$

where ρ_{12} = light-time-correct range for leg from 1 to 2

t_1 = transmit time for tracking signal (to be determined)

t_2 = receive time for tracking signal (known)

$\vec{r}_1$ = known position of transmitter (known function during the iterative process)

$\vec{r}_2$ = known position of receiver (fixed during the iterative process)

c = vacuum speed of light (see Table 3-1)

Equations (3-7) and (3-8) are solved iteratively for ρ_{12} and t_1, as follows. Using the last value of ρ_{12}, Equation (3-7) is used first to evaluate t_1. Then Equation (3-8) is used to evaluate a better value of ρ_{12}. This two-step procedure is repeated until the change in t_1 is less than 10^{-8} seconds. The iterations are initialized with the geometric range. After all four legs have been computed, the range measurement is computed as one-half of the sum of the values for the four legs.

The geometric partial derivatives, Equations (3-6), are adequate and are used even when a light-time-correct range is computed.

3.2.1.3 Preprocessing of Real TDRSS Range Tracking Data

Two corrections for the actual relay range measurements in the TDRSS tracking data must be treated. These are the transponder delay and the range ambiguity.

Table 3-1. Physical Constants

Quantity	Numerical Value
Speed of Light, c	2.99792458×10^5 kilometers/second
Gravitation constant of Earth, GM	398600.47 kilometers^3/second^2
Earth sidereal rotation rate, ω_E	6.300388098445825 radians/day

The transponder delay correction is sometimes already included in the values. If so, it is removed by retrieving the value, $\Delta\rho$, available in the TDRSS data and using the formula

$$\rho_{\text{uncorrected}} = \rho_{\text{corrected}} + \Delta\rho \qquad (3\text{-}9)$$

The stored value of the transponder delay can then be included or not in subsequent early orbit testing.

The TDRSS observations are of the ambiguous range. The unambiguous range values are computed by using the algorithm in Reference 6. This algorithm requires the use of a sufficiently good nominal spacecraft state vector to derive the correct unambiguous range.

3.2.2 TDRSS RELAY DOPPLER MEASUREMENTS

3.2.2.1 Geometric Range-Rate and Partial Derivatives

The geometric model for the single-relay range-rate measurement is as follows:

$$\dot{\rho} = \frac{(\vec{r} - \vec{R}_{TDRS})}{|\vec{r} - \vec{R}_{TDRS}|} \cdot (\vec{v} - \vec{V}_{TDRS}) + \frac{(\vec{R}_{TDRS} - \vec{R}_{WS})}{|\vec{R}_{TDRS} - \vec{R}_{WS}|} \cdot (\vec{V}_{TDRS} - \vec{V}_{WS}) \qquad (3\text{-}10)$$

where

$\dot{\rho}$ = geometric range-rate

$\vec{r}$, $\vec{v}$ = position, velocity vectors of the target spacecraft at time t (see Section 3.2.4)

$\vec{R}_{TDRS}$, $\vec{V}_{TDRS}$ = position, velocity vectors of the TDRS at time t (see Section 3.2.4)

$\vec{R}_{WS}$, $\vec{V}_{WS}$ = position, velocity vectors of the White Sands ground station at time t (see Section 3.2.5)

Assuming that $\vec{R}_{WS}$, $\vec{V}_{WS}$, $\vec{R}_{TDRS}$, and $\vec{V}_{TDRS}$ remain fixed throughout the calculation, the local partial derivatives needed for Equations (3-4) are given by

$$\frac{\partial \rho}{\partial y_i} = \frac{(v_i - V_{TDRS,i})}{|\vec{r} - \vec{R}_{TDRS}|} - \frac{(\vec{v} - \vec{V}_{TDRS}) \cdot (\vec{r} - \vec{R}_{TDRS})}{|\vec{r} - \vec{R}_{TDRS}|^3} (r_i - R_{TDRS,i}), \quad i = 1, 2, 3 \tag{3-11}$$

$$\frac{\partial \rho}{\partial y_i} = \frac{(r_i - R_{TDRS,i})}{|\vec{r} - \vec{R}_{TDRS}|}, \quad i = 4, 5, 6$$

3.2.2.2 Modeling of the Light-Time-Correct Doppler Measurement

In the light-time-correct Doppler modeling, it is assumed that during preprocessing the Doppler measurement has been converted to an equivalent averaged range-rate value and that the pilot tone effect of the relative motion of the TDRS itself has been removed from the measurement value, which is discussed in the next subsection. Then, this average range rate is modeled as follows:

$$\dot{\rho} = \frac{\rho(t_R) - \rho(t_R - \Delta t)}{\Delta t}$$

where $\dot{\rho}$ = modeled value of the averaged range rate

t_R = observation time tag at the end of the Doppler count interval

Δt = length of the Doppler count averaging interval

$\rho(t)$ = light-time-correct range, calculated according to Section 3.2.1.2, for the signal received at time t

Thus, for each Doppler measurement, it is necessary to iteratively compute eight legs for the tracking signal.

The geometric partial derivatives, given in Equations (3-11), are used with adequate accuracy with the light-time-correct Doppler model.

3.2.2.3 Preprocessing of Real TDRSS Doppler Tracking Data

Preprocessing of the TDRSS Doppler tracking measurement is based upon the formulas given in Reference 5 and also in Section 5.5.3 of Reference 6. Thus, the averaged range-rate measurement, $\dot{\rho}$, is computed from the measured Doppler frequency F_d using

$$\dot{\rho} = - \frac{cF_d}{F_{ref}} + \frac{bF_p}{F_{ref}} \cdot \frac{\Delta R_s}{\Delta t}$$

where F_{ref} = effective user transmit frequency (specified in the TDRSS tracking messages)

bF_p = return TDRS frequency translation (determined by the downlink channel as specified in Reference 6)

Δt = Doppler count interval

ΔR_s = range change of the TDRS during the Doppler count interval

c = vacuum speed of light

ΔR_s is modeled, using the iterative light time calculation, as

$$\Delta R_s = \rho_s(t) - \rho_s(t - \Delta t)$$

where $\rho_s(t)$ is the short range (White Sands to TDRS and back) tagged at the end of the Doppler count interval and $\rho_s(t - \Delta t)$ is the short range tagged at the start of the Doppler count interval.

3.2.3 TDRS ANTENNA BEAM ANGLES

3.2.3.1 Geometric Antenna Beam Angles and Partial Derivatives

The formulas for modeling the TDRS antenna beam azimuth, A, and elevation, E, are as follows:

$$\tan A = \frac{X3}{(-X1)} \tag{3-12}$$

$$\tan E = \frac{X2}{(X1^2 + X3^2)^{1/2}} \tag{3-13}$$

where X1, X2, and X3 are defined by

$$X1 = (\vec{r} - \vec{R}_{TDRS}) \bullet \vec{e}_1$$

$$X2 = (\vec{r} - \vec{R}_{TDRS}) \bullet \vec{e}_2$$

$$X3 = (\vec{r} - \vec{R}_{TDRS}) \bullet \vec{e}_3$$

and the unit vectors $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ are defined by

$$\vec{e}_1 = \frac{\vec{R}_{TDRS}}{|\vec{R}_{TDRS}|}$$

$$\vec{e}_3 = \frac{\vec{R}_{TDRS} \times \vec{V}_{TDRS}}{|\vec{R}_{TDRS} \times \vec{V}_{TDRS}|} \tag{3-14}$$

$$\vec{e}_2 = \vec{e}_3 \times \vec{e}_1$$

In these equations, $\vec{r}$, $\vec{v}$, $\vec{R}_{TDRS}$, and $\vec{V}_{TDRS}$ are as defined for Equation (3-10). These formulas are similar to those given in Section 5.5.4 of Reference 6, although the notation is different.

In terms of the TDRS-to-target components, X1, X2, and X3, the local partial derivatives are given by

$$\frac{\partial A}{\partial y_i} = \cos^2 A \left[\frac{e_{2,i}}{(-X1)} + \frac{X3}{(X1)^2} y_i \right] \qquad i = 1, 2, 3$$

$$\frac{\partial A}{\partial y_i} = 0, \quad i = 4, 5, 6 \qquad (3\text{-}15)$$

and

$$\frac{\partial E}{\partial y_i} = \cos^2 E \left[\frac{e_{3,i}}{(X1^2 + X3^2)^{1/2}} - \frac{X2(X1e_{1,i} + X3e_{3,i})}{(X1^2 + X3^2)^{3/2}} \right]$$

$$i = 1, 2, 3 \qquad (3\text{-}16)$$

$$\frac{\partial E}{\partial y_i} = 0, \quad i = 4, 5, 6$$

Because TDRS beam angle values are not actual measurements, a light-time-correct beam angle model is not formulated, since the additional accuracy would not be warranted.

3.2.3.2 Preprocessing of Real TDRS Antenna Beam Angles

The direction cosines X_R, Y_R, and Z_R are retrieved from the TDRSS tracking observation records (words 6, 7, and 8). These direction cosines define the direction (return link) from the TDRS to the target with respect to TDRS reference coordinates, and represent the antenna beam angles after correction for the nonnominal TDRS attitude. The direction cosines X_R, Y_R, and Z_R differ from X1, X2, and X3 in the previous section and are defined as follows:

X_R = projection of the TDRS-to-spacecraft unit vector onto the direction in the TDRS orbit plane (eastward) that is perpendicular to the direction from the TDRS to the center of the Earth

Y_R = projection of the TDRS-to-spacecraft unit vector onto the direction normal to the TDRS orbit plan (south)

Z_R = projection of the TDRS-to-spacecraft unit vector onto the direction from the TDRS to the center of Earth

X_R, Y_R, and Z_R are converted to azimuth and elevation by means of the formulas

$$A = \tan^{-1}\left(\frac{X_R}{Z_R}\right)$$

$$E = \tan^{-1}\left(\frac{-Y_R}{\sqrt{X_R^2 + Z_R^2}}\right)$$

A and E are the quantities to be used in early orbit determination.

3.2.4 ORBIT PROPAGATION

Propagation of trajectories from initial states at the reference time to arbitrary times, for which state vectors are required for measurement modeling, was required for both the TDRS and the target spacecraft. These propagations provide the vectors $\vec{r}$, $\vec{v}$, $\vec{R}_{TDRS}$, and $\vec{V}_{TDRS}$, which appear in Sections 3.2.1 through 3.2.3.

For the TDRS, two-body propagation was used in all cases. (See Section 5.7.3 in Reference 6, and Reference 12 of Section 5 in Reference 6 for descriptions of the closed-form, two-body propagator that was used.) The value of the Earth gravitational constant is listed in Table 3-1. The use of the two-body approximation for the TDRS trajectory produces propagation errors of 0.1 to 0.2 kilometers during a 100-minute propagation interval, when compared with a Cowell propagator with a precise force model (see Table 3-2). For the particular case described in the table, there is some cancellation occurring between lunar-solar and Earth gravity

Table 3-2. Differences Between TDRS Ephemerides

Position Difference Component	Maximum Position Difference Over the 100-Minute Comparison Interval Between Two Ephemerides[1] (Km)			
	Solar Radiation Force Omitted	Sun and Moon Gravity Omitted	Gravity Harmonics Omitted	All Three Perturbations Omitted
Radial	0.003	0.153	0.148	0.008
Cross-Track	0.0002	0.102	0.002	0.104
Along-Track	0.0008	0.049	0.044	0.004
Total	0.003	0.190	0.154	0.105

[1]The ephemerides compared both beginning from the same initial state on March 14, 1984, at 0^h. One ephemeris includes all perturbations; the other ephemeris in the comparison omits one or more of the perturbations, as indicated in the table.

effects; at different times the errors may add rather than cancel. In the worst case, then, the error may be as large as 300 meters. This error can often be ignored in early orbit determination.

The error attributable to the two-body approximation in the determined orbit of the target spacecraft will depend, of course, on the orbit type. For a low-altitude spacecraft (for example, Landsat-4), the error over one revolution is about 50 to 100 kilometers. This is examined in more detail in Section 4.1. However, this error is sufficiently large that it needs to be eliminated in early orbit determination. In this study, a Brouwer-Lyddane propagator was selected to accomplish this. This propagator is described in Section 5.10 of Reference 7. Two-body state partial derivatives were used with the Brouwer-Lyddane propagator, without any resultant numerical difficulties in the curve-following algorithm described in Section 3.4.

3.2.5 GROUND STATION POSITION AND VELOCITY

Irregularities in the Earth's rotation rate are ignored, and the position and velocity of the White Sands ground station are modeled with the following formulae:

$$\begin{aligned} \vec{R}_{WS} &= T \bullet \vec{R}_o \\ \vec{V}_{WS} &= \frac{d}{dt} T \bullet \vec{R}_o \end{aligned} \qquad (3\text{-}17)$$

where R_o is the Earth-fixed station location and T(t) is the 3x3 rotation matrix defined by

$$T(t) = \begin{pmatrix} \cos \alpha(t) & -\sin \alpha(t) & 0 \\ +\sin \alpha(t) & \cos \alpha(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (3\text{-}18)$$

The variable t is the number of seconds of Universal Time, Coordinated (UTC) elapsed from the reference time, and the angle α is computed from

$$\alpha = \alpha_o + \omega_E t \tag{3-19}$$

The constant α_o is the Greenwich Hour Angle at the reference time. The numerical value of the Earth rotation rate, ω_E, is listed in Table 3-1.

Table 3-3 provides the Earth-fixed station location of the antenna WH2K, which collected the tracking data for the Landsat-4 tests in Section 4.1. The coordinates given have been adjusted to WGS-72 (World Geodetic System, 1972).

3.3 SIMPLE EXAMPLE

A highly simplified orbit determination problem is solved here to illustrate the meaning of Equations (3-3). In this example, the orbit state is one-dimensional, leading to an explicit formula for the solution curve.

The tracking geometry is shown in Figure 3-4. It is assumed that the TDRS lies in the orbit plane. The radius of the circular orbit is assumed to be fixed, and the single component of the state vector is the cartesian coordinate x. The solution state is denoted by x^1, and the a priori estimate is denoted, by x^0. For definiteness, it is assumed that $x^0 > x^1$ and that both of these states lie in the first quadrant of the angle Θ. The measurement is assumed to be the square of the geometrical range ρ.

From Figure 3-4, the following geometric relations hold:

$$o^1 = a^2 + R^2 + 2aR \left[1 - \left(\frac{x^1}{a}\right)^2\right]^{1/2}$$

Table 3-3. Earth-Fixed Coordinates for the Antenna WH2K at White Sands

Cartesian Component	Geocentric Coordinates (km)
X	-1539.404223
Y	-5160.963938
Z	+3408.172440

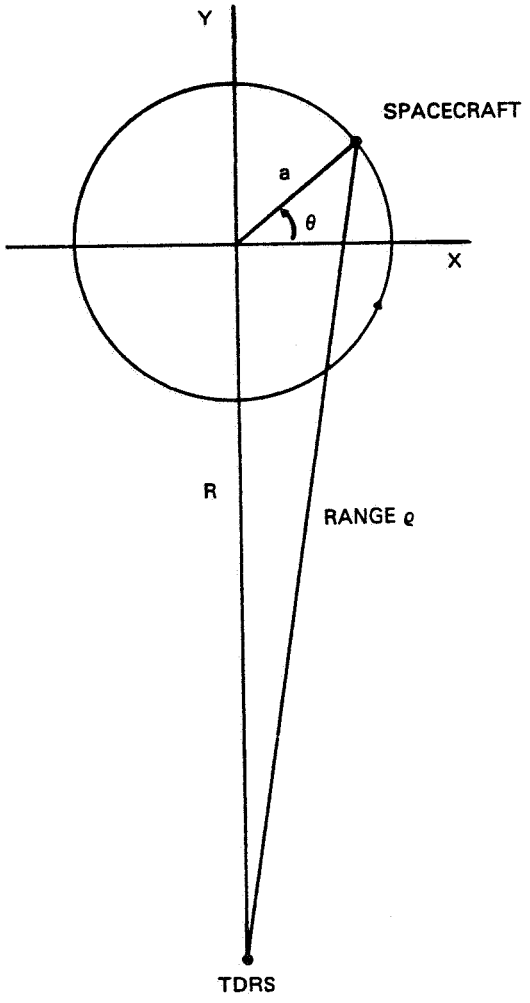


Figure 3-4. Tracking Configuration for the One-Dimensional Example of the Homotopy Method

$$O^0 = a^2 + R^2 + 2aR\left[1 - \left(\frac{x^0}{a}\right)^2\right]^{1/2}$$

$$C = a^2 + R^2 + 2aR(\pm 1)\left[1 - \left(\frac{x}{a}\right)^2\right]^{1/2}$$

where

$$\sin\theta = \pm\left[1 - \left(\frac{x}{a}\right)^2\right]^{1/2}$$

has been used.

O^1 is the given measurement, while O^0 is the modeled measurement value for the a priori estimate x^0.

Substitution of these relations into Equations (3-3) yields the equation for the solution curve in the λ-x plane:

$$\left\{\left[1 - \left(\frac{x^1}{a}\right)^2\right]^{1/2} - \left[1 - \left(\frac{x^0}{a}\right)^2\right]^{1/2}\right\} + \left[1 - \left(\frac{x^0}{a}\right)^2\right]^{1/2} = \pm\left[1 - \left(\frac{x}{a}\right)^2\right]^{1/2}$$

or, equivalently,

$$\left(\frac{\lambda + \frac{B}{A}}{\frac{1}{A}}\right)^2 + \left(\frac{x}{a}\right)^2 = 1 \qquad (3\text{-}20)$$

where

$$A \equiv \left[1 - \left(\frac{x^1}{a}\right)^2\right]^{1/2} - \left[1 - \left(\frac{x^0}{a}\right)^2\right]^{1/2}$$

$$B \equiv \left[1 - \left(\frac{x^0}{a}\right)^2\right]^{1/2}$$

The solution curve described by Equation (3-20) is an ellipse, centered at the point (-B/A, 0), having a semihorizontal axis of length 1/A and a semivertical axis of length a (see Figure 3-5). At each value of λ corresponding to points on the ellipse, except for the two critical points at x = 0, there are two orbit states. This duplicity physically corresponds to the fact that a circle of radius ρ can intersect the orbit at two distinct points. At each of the two critical points, the two orbit states coalesce to form a single state, and the observation derivative, dC/dx, is precisely equal to zero at these points. Orbit solution states do not exist for values of λ outside the range covered by the solution curve, and the intersection points of the solution curve with the vertical line, λ = 1, identify all of the orbit solutions that exist for the given measurement, O^1. (Actually, the variable used to describe the orbit state in this example, x, is not really a good choice because for each value of x there are, physically, two orbit states. In Figure 3-5 this corresponds to the fact that there are two points on the curve at a given value of x. This choice was made in order to get a closed curve, analogous to the closed curves obtained in the full six-dimensional problem. If a good state variable (for example, θ) is used instead, the solution curve is not closed, but repeats with a period 2π. The corresponding λ-θ solution curve is given in Reference 8.)

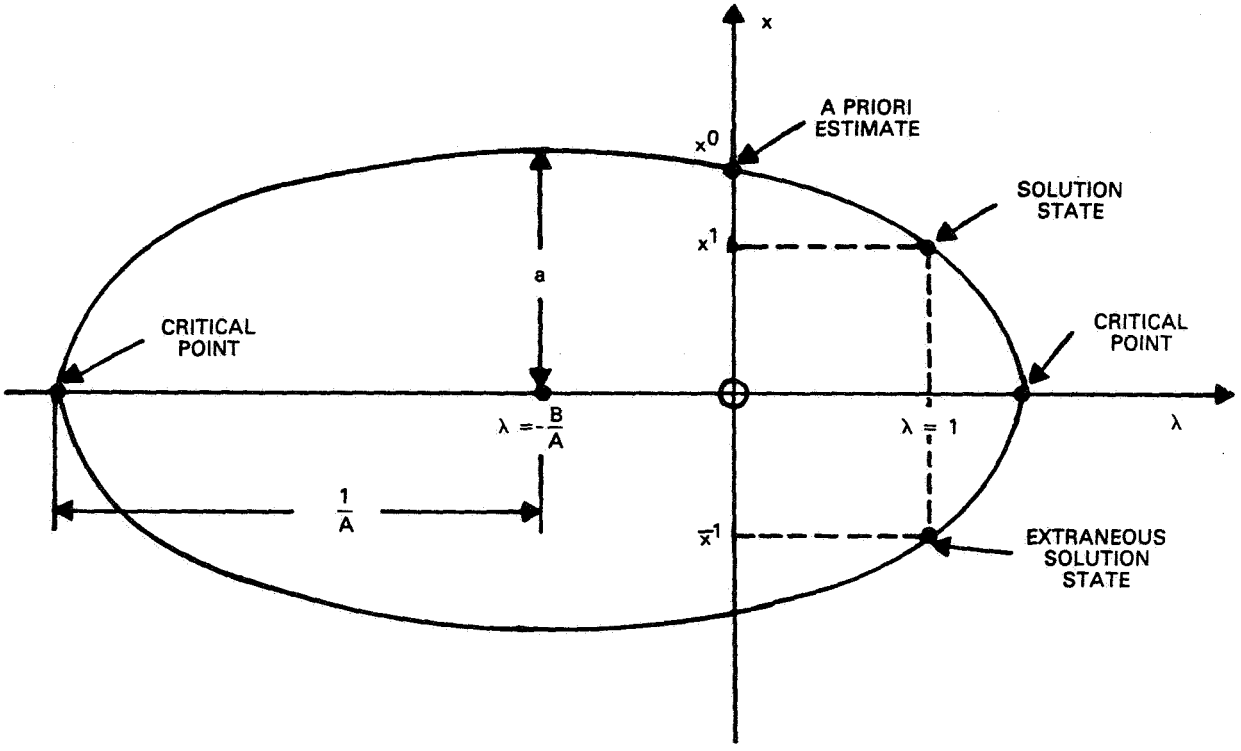


Figure 3-5. Solution Curve for the One-Dimensional Example of the Homotopy Method

3.4 RELATION TO THE NEWTON-RAPHSON METHOD

The particular formulation of the homotopy method described in Section 3.1 for orbit determination can be considered to be a generalization of the standard Newton-Raphson method for solving systems of equations. Given an a priori estimate, $\vec{x}^0$, the Newton-Raphson method computes the correction, δx, using the equation

$$\delta x = -B^{-1}(x^0) \cdot A(x^0) \tag{3-21}$$

where $B(x^0)$ is the matrix of partial derivatives at x^0,

$$-B(x^0) \equiv \begin{bmatrix} \frac{\partial C_1}{\partial x_1} & \cdots & \frac{\partial C_1}{\partial x_6} \\ \vdots & & \\ \frac{\partial C_6}{\partial x_1} & \cdots & \frac{\partial C_6}{\partial x_6} \end{bmatrix}_{\vec{x}^0}$$

and A is the column vector of residuals evaluated at x^0,

$$A(x^0) \equiv \begin{bmatrix} O_1^1 - C_1(\vec{x}^0) \\ \vdots \\ O_6^1 - C_6(\vec{x}^0) \end{bmatrix}$$

The homotopy method defined by Equations (3-3) will be put into a form that will enable comparison with Equations (3-21).

Since the solution curves are almost always smooth, the ordinary arc length, s, can be introduced as the curve parameter. By definition, changes in the arc length are related to changes in λ and x through the formula

$$\left(\frac{d\lambda}{ds}\right)^2 + \left(\frac{d\vec{x}}{ds} \cdot \frac{d\vec{x}}{ds}\right) = 1 \qquad (3\text{-}22)$$

Next, differentiation of each side of Equation (3-3) with respect to s yields

$$\left(o_i^1 - o_i^0\right) \frac{d\lambda}{ds} - \sum_j \frac{\partial C_i}{\partial x_j} \frac{dx_j}{ds} = 0, \qquad i = 1, \ldots, 6$$

or, equivalently,

$$\vec{A}(\vec{x}^0) \frac{d\lambda}{ds} + B(x) \cdot \frac{d\vec{x}}{ds} = 0 \qquad (3\text{-}23)$$

Finally, solving Equations (3-22) and (3-23) for $d\lambda/ds$ and dx/ds yields

$$\begin{aligned} \frac{d\lambda}{ds} &= + \frac{1}{G(\vec{x})} \\ \frac{d\vec{x}}{ds} &= - \frac{B^{-1}(\vec{x})\, A(\vec{x}^0)}{G(x)} \end{aligned} \qquad (3\text{-}24)$$

where

$$G(\vec{x}) = \pm \left[1 + A^T(\vec{x}^0)\, B^{-T}(\vec{x})\, B^{-1}(\vec{x})\, A(\vec{x}^0)\right]^{1/2} \qquad (3\text{-}25)$$

These are the differential equations satisfied by the solution curve. The double choice in sign corresponds to following the curve in either of the two possible directions.

The Newton-Raphson method consists of integrating Equations (3-24) with the simple Euler method and a step Δs that extends from $\lambda = 0$ to $\lambda = +1$, approximately. From Equation (3-24), this step has length

$$\Delta s = G(\vec{x}^0) \cdot 1$$

(+ sign is chosen for G) and the corresponding change in x is given by

$$\Delta x = - \frac{B^{-1}(\vec{x}^0)\ A(\vec{x}^0)}{G(\vec{x}^0)} \Delta s \qquad (3\text{-}26)$$

or

$$\vec{\Delta x} = -B^{-1}(\vec{x}^0)\ A(\vec{x}^0)$$

which is precisely the change given by Equation (3-21) for the Newton-Raphson method.

Thus, one iteration of the Newton-Raphson method corresponds to following the solution curve from $\lambda = 0$ to $\lambda = 1$ using the straight line approximation that is tangent to the solution curve at $\lambda = 0$. Each Newton-Raphson iteration, in turn, constructs such a line tangent to a new solution curve. Returning to the simple example in Section 3.3, as $x^0 \rightarrow x^1$, then $1/A \rightarrow \infty$, the ellipse becomes horizontally elongated, and the segment of the solution curve between $\lambda = 0$ and $\lambda = 1$ becomes straighter, which leads to

better convergence of the Newton-Raphson method. It is expected, however, that, if the initial solution curve is strongly curved between $\lambda = 0$ and $\lambda = 1$, then the Newton-Raphson method will probably not converge (see Figure 3-6).

3.5 NUMERICAL ALGORITHM FOR FOLLOWING SOLUTION CURVES

The algorithm of this section permits the solution states that are located on the same component of the solution curve as the a priori state to be determined up to machine precision. The more general case in which the desired solution state and the a priori state lie on disjoint components is considered in Section 6.2. The algorithm for the restricted case forms the major part of the general, more complete algorithm in Section 6.2.

The algorithm was found to perform reliably in hundreds of test cases. However, in some of the component areas, minor improvements are suggested here, through which the efficiency (that is, the relative amount of computation per solution) might be improved in a subsequent version of the algorithm. Although developed independently, the algorithm given here is similar to that described in Reference 9.

The algorithm assumes scaled cartesian variables for orbit states. The length scale is the radius of the Earth, and the velocity scale is the circular speed in an orbit at one Earth radius. The homotopy parameter λ is not scaled, although such scaling would be convenient, because it is not now clear how the approximate size of a solution loop can be estimated. Scaling of λ must await a deeper understanding of the global nature of the solution curves.

The algorithm follows the solution curve (defined by an a priori estimate, the six given observations, and the trajectory and observation models) in seven-dimensional

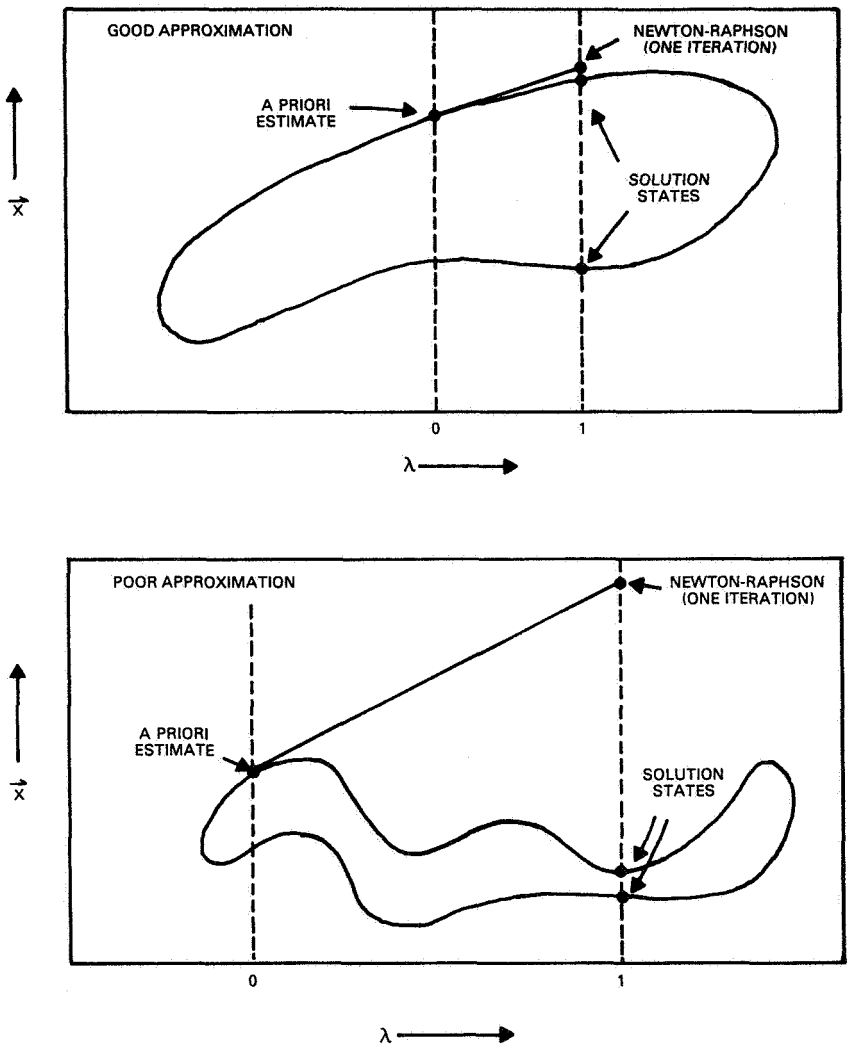


Figure 3-6. One-Step Euler Method for Following a Solution Curve (Schematic)

λ-x space by constructing a sequence of points that lie exactly on the curve. At any intermediate stage of the process, a succeeding point is determined with a two-step, predictor-corrector method. An estimate of the succeeding point is computed using a Lagrange polynomial fit through a selected number of back points (predictor) in each of the seven coordinates, λ, x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$. This estimate is then refined to the specified precision using the Newton-Raphson method such that successive iterative corrections are constrained to lie in a six-dimensional hyperplane that is approximately perpendicular to the curve (corrector). This constraint avoids Jacobian ill-conditioning problems that can arise if, instead, the iterations were performed at fixed λ. The predictor-corrector technique is schematically illustrated by Figure 3-7.

As the points on the solution curve are successively computed, the algorithm must check for solution states at $\lambda = 1$ and refine and store these solution states. The algorithm must also check for possible return to the a priori state. Optionally, the algorithm checks for and identifies critical points (local extrema in λ (s)). This is necessary only for the multiloop algorithm in Section 6.2, but is described below for completeness.

The major steps in the algorithm are listed below. These steps are detailed in Sections 3.5.1 through 3.5.9. Section 3.5.10 indicates extreme conditions although they are unlikely to arise in practice) under which the algorithm can (numerically) fail.

Step 1. Bootstrap Starter (Section 3.5.1). A special sub-algorithm is required at the start because only one point on the solution curve, the specified

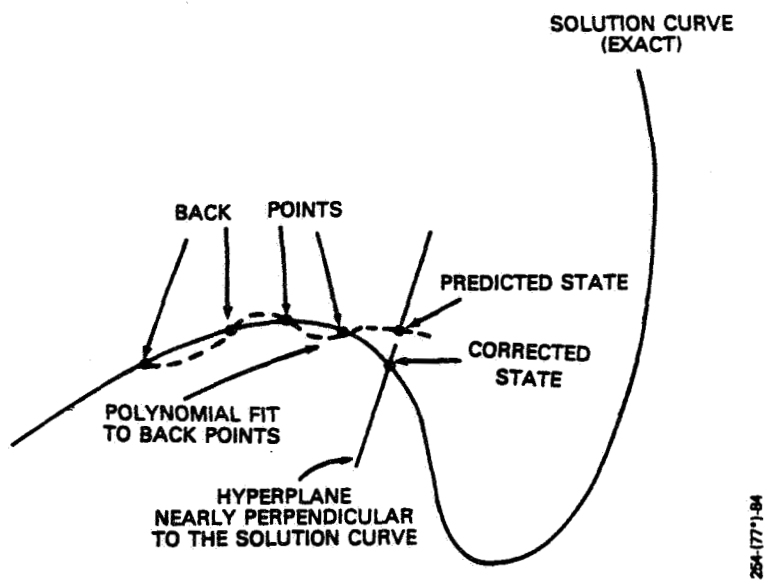


Figure 3-7. Predictor-Corrector Technique for Following a Solution Curve

a priori state, is known, while the direction of the curve is completely unknown. This starter determines one additional point on the curve.

Step 2. Correction of Arc Length (Section 3.5.2). Step 2 is the first step of the main loop of the algorithm. At the conclusion of the previous corrector step, the arc length change, $\Delta s'$, is slightly incorrect because of the corrector process. Gaussian quadrature is used to compute a refined numerical value, Δs.

Step 3. Collection of Output States (Section 3.5.3). The predictor-corrector steps may have followed the solution curve across $\lambda = 1$. If necessary, the $\lambda = 1$ state is iteratively estimated and refined until it is determined to within the specified tolerance.

Step 4. Collection of Critical Points (Section 3.5.4). The predictor-corrector steps may have followed the solution curve through a local extremum in λ. In this step, such a critical point is first estimated and then refined to within the specified tolerance.

Step 5. Termination (Section 3.5.5). The predictor-corrector steps may have followed the solution curve back to and past the starting point. This condition is checked, and the calculation is terminated if it occurred.

Step 6. Step Size Selection (Section 3.5.6). The computation of the next point on the solution curve begins with Step 6. A preliminary value for the arc length change, $\Delta s'$, from the last back point to the next curve point is selected on the basis of properties of the last back point calculation.

Step 7. Prediction of New Curve Point (Section 3.5.7). Using Lagrange interpolation, a polynomial fit to the last N backpoints is performed. This polynomial is then used to evaluate the estimate of the new curve point at a position that is advanced in arc length by the selected amount, $\Delta s'$.

Step 8. Correction of New Curve Point (Section 3.5.8). Using the Newton-Raphson method in a hyperplane perpendicular to the extrapolating polynomial at the predicted point, the predicted state is iteratively refined until Equations (3-3) are satisfied to within the specified tolerance.

Step 9. Monitoring of New Curve Points (Section 3.5.9). If the corrector iterations in Step 8 do not converge, or if the direction of the curve tangent changes by too large an amount from the last backpoint to the new curve point, then the new curve point is discarded, the step size, $\Delta s'$, is reduced, and steps 7 and 8 are repeated. If, on the other hand, the new curve point is acceptable, the algorithm advances along the curve by one unit and then returns to the start of the main algorithm loop at step 2.

3.5.1 BOOTSTRAP STARTER

The a priori state is denoted by $u^0 = (0, x^0, y^0, z^0, \dot{x}^0, \dot{y}^0, \dot{z}^0)^T$. By definition, this state lies on the solution curve, and the observations, O_i^0, $i = 1, \ldots, 6$, are simulated on the basis of this state prior to the execution of this starting procedure.

The bootstrap starter searches over seven orthogonal directions, attempting to find a second state on the solution curve. In turn, trial predicted states are generated which

lie in the λ-, x-, y-, z-, x-, y-, and z-directions with respect to the a priori state, and at a selected distance $\Delta s'$ from the a priori state. Thus, these seven predicted states are

$$P_1 = \begin{pmatrix} \pm\Delta s' \\ x^0 \\ y^0 \\ z^0 \\ x^0 \\ y^0 \\ z^0 \end{pmatrix} \qquad P_2 = \begin{pmatrix} 0 \\ x^0 \pm\Delta s' \\ y^0 \\ z^0 \\ x^0 \\ y^0 \\ z^0 \end{pmatrix} \qquad \ldots \; P_7 = \begin{pmatrix} 0 \\ x^0 \\ y^0 \\ z^0 \\ x^0 \\ y^0 \\ z^0 \pm\Delta s' \end{pmatrix}$$

The sign choice controls the directional sense in which the algorithm follows the curve.

Using these seven trial predicted states in turn, the starter attempts Newton-Raphson iterative refinement (described in Section 3.5.8). If the refinement is successful for any particular predicted state, then a second point on the solution curve has been determined and the starter terminates. If the Newton-Raphson iterations do not converge, the next predicted state is attempted.

If all seven predicted states are unsuccessful, then the attempted step, $\Delta s'$, is reduced by a factor of two, and the entire process (seven predicted states) is repeated. The factor-of-two reduction can be repeated up to a specified maximum number of times. For a well-posed orbit determination problem, starter failure only occurs when there is an error in the program coding or in the data because, as $\Delta s'$ is reduced, the predicted state becomes increasingly accurate, and the Newton-Raphson procedure must converge at some sufficiently small value of the initial step, $\Delta s'$.

This bootstrap starter can also be used to begin calculation at some arbitrary initial state that is not the a priori state, perhaps a state on a partially computed curve. The procedure is analogous, except that the initial value of λ is not zero but some other value.

3.5.2 CORRECTION OF ARC LENGTH

At the start of this subalgorithm, a number of curve points, $\vec{u}_1$, $\vec{u}_2$, ..., $\vec{u}_N$ (the subscripts increase backward along the curve), are known, where N is the selected order. The points correspond to arc length values s_1', s_2, ..., s_N, where, again, the subscripts increase backward along the curve. The value of s_1' is provisional, and it is refined by this subalgorithm.

The corrected value of the current arc length, s_1, is calculated using

$$s_1 = s_2 + \int_{s_2}^{s_1'} \left[\frac{\vec{du}}{d\sigma} \cdot \frac{\vec{du}}{d\sigma} \right]^{1/2} d\sigma \qquad (3\text{-}27)$$

where $u(\sigma) = (\lambda(\sigma), x(\sigma), y(\sigma), z(\sigma), \dot{x}(\sigma), \dot{y}(\sigma), \dot{z}(\sigma))^T$.

The tangent vector, $du/d\sigma$, is evaluated as the derivative of the N-point Lagrange interpolating polynomial (see Section 3.5.7), and the integral is numerically calculated with standard N-point Gaussian quadrature (see Reference 10).

The purpose of this step is to keep the arc length variable accurate so that the solution curve tangent vector, required in Section 3.5.9, is accurate. Because the differences between s_1' and s_1 have been observed to be rather small in practice, future work may eventually prove that this step is unnecessary.

3.5.3 COLLECTION OF OUTPUT STATES

At the start of this subalgorithm, two curve points, $\vec{u}_1$ and $\vec{u}_2$, are known, along with the associated arc lengths, s_1 and s_2. The objective is to determine the precise points, if any, at which the solution curve crosses the $\lambda = 1$ hyperplane. It should be noted that polynomial interpolation does not generally perform this task with sufficient precision unless the curve steps, Δs, are inefficiently small. Furthermore, Newton-Raphson iteration at fixed λ, initiated with such an interpolated state, will occasionally fail if the curve is close to being parallel to the $\lambda = 1$ hyperplane. Therefore, a more careful, and also more reliable, approach has been developed and is described here.

First, to save time, if both $\vec{u}_1$ and $\vec{u}_2$ are far from the $\lambda = 1$ hyperplane, it is assumed that no solution states were crossed during the step and, in this case, the subalgorithm is terminated.

If, on the other hand, either $\vec{u}_1$ or $\vec{u}_2$ is not far from $\lambda = 1$, then the number and approximate locations of solution states between s_2 and s_1 are first determined. This is done by computing interpolated states, $\vec{u}'_1$, $\vec{u}'_2$, ..., u'_M at M values of arc length uniformly distributed between s_2 and s_1. M is typically 50, and the N-point Lagrange interpolating formula is used for this procedure (see Section 3.5.7). The number and approximate locations of the solution states are then determined from the signs of $1-\lambda'_j$ in the sequence of interpolated states. For each solution state, there will be two nearby interpolated states, $\vec{u}'_k$ and $\vec{u}'_{k+1}$ that straddle the solution state. Figure 3-8 shows cases having one and two solution states.

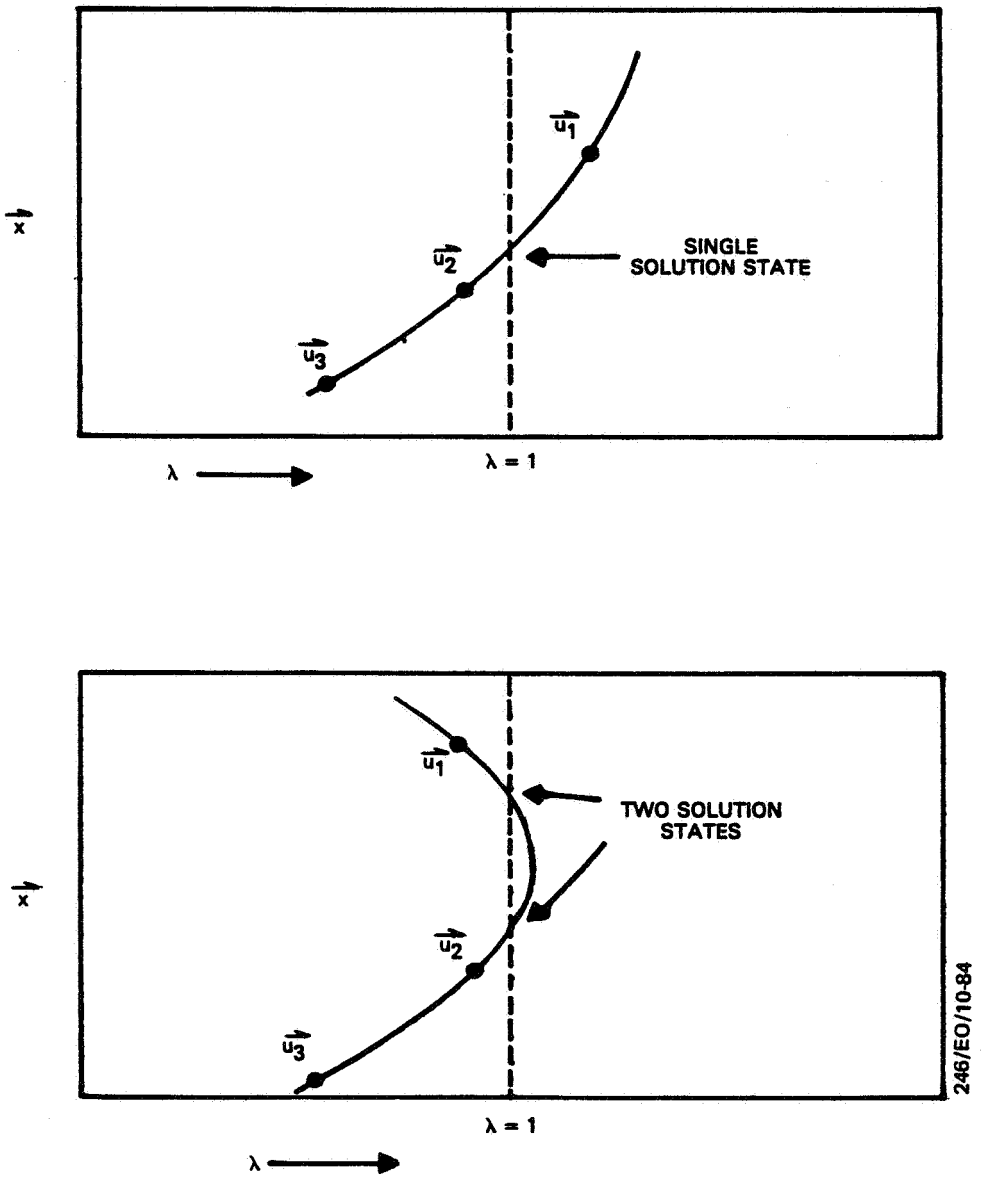


Figure 3-8. Single and Double Solution States Crossed During One Curve-Following Step

Next, the solution state refinement process begins; an iterative procedure is used to refine each of the candidate solutions, in turn. This procedure consists of the following steps:

Step 1. The two interpolated states, $\vec{u}'_k$ and $\vec{u}'_{k+1}$, are each refined, using the Newton-Raphson method (Section 3.5.8). This yields $\vec{u}_k$ and $\vec{u}_{k+1}$, which lie on the solution curve.

Step 2. The λ components of $\vec{u}_k$ and $\vec{u}_{k+1}$ are generally not equal to 1. A linearly interpolated state, $\vec{u}_A$, at $\lambda = 1$ is computed using

$$\vec{u}'_A = \vec{u}_k + \frac{\left(1 - \lambda_k\right)}{\left(\lambda_{k+1} - \lambda_k\right)} \left(\vec{u}_{k+1} - \vec{u}_k\right) \qquad (3\text{-}28)$$

Step 3. $\vec{u}'_A$ is refined using the Newton-Raphson method (Section 3.5.8) to derive $\vec{u}_A$, which lies on the solution curve.

Step 4. If the λ component of $\vec{u}_A$ is equal to 1, to within the specified tolerance, then the refinement process is complete for this solution, and the process begins again with the next candidate solution, if any. On the other hand, if λ_A is not sufficiently close to 1, then the refinement process continues with step 5.

Step 5. The value of $\vec{u}_k$ is set to either the value of $\vec{u}_k$ or $\vec{u}_{k+1}$, whichever has a λ component that is closer to one, and $\vec{u}_{k+1}$ is set to $\vec{u}_A$. Then, the process returns to step 2 for another iteration.

Usually, this refinement process is completed with two or three iterations. The refinement process may occasionally have an abnormal termination due to nonconvergence. This

can happen if the interpolating polynomial crosses $\lambda = 1$, but the actual solution curve does not. If a nonconvergence condition occurs, then the last curve point, $\vec{u}_1$, should be discarded; a smaller step size, $\Delta s'$, selected; and a new curve point computed.

3.5.4 COLLECTION OF CRITICAL POINTS

To reliably and efficiently evaluate critical points, the curve-following algorithm should continually select steps, Δs, that are about as small as, or smaller than, the local radius of curvature of the solution curve. The subalgorithm described in Section 3.5.9 is intended to accomplish that.

At the start of the critical point subalgorithm, three curve points, $\vec{u}_1$, $\vec{u}_2$, and $\vec{u}_3$, and the associated arc lengths, s_1, s_2, and s_3, are available. The objective is to determine if a local extremum in $\lambda(s)$ exists in the solution curve between s_3 and s_1, and, if so, to evaluate the coodinates precisely.

First, the subalgorithm checks for the following conditions:

$$\lambda_2 > \lambda_1 \text{ and } \lambda_2 > \lambda_3 \tag{3-29}$$

$$\lambda_2 < \lambda_1 \text{ and } \lambda_2 < \lambda_3 \tag{3-30}$$

If neither of these two conditions is satisfied, then it is assumed that a local extremum is not contained between s_3 and s_1, and the subalgorithm terminates. On the other hand, if either of these two conditions is satisfied, then it is assumed that exactly one local extremum exists between s_3 and s_1, and the subalgorithm continues.

An approximate location of the critical point is first determined. A sequence of interpolated derivatives,

$$\left.\frac{d\lambda'}{ds}\right|_1, \left.\frac{d\lambda'}{ds}\right|_2, \ldots, \left.\frac{d\lambda'}{ds}\right|_M$$

is computed for M values of the arc length that are uniformly distributed between s_3 and s_1. M is typically 100, and the derivative of the N-point Lagrange interpolating polynomial is used (see Section 3.5.7). The approximate location of the extremum is determined from the sequence of signs in the sequence of interpolated derivatives. The arc length value at the position of the extremum in the interpolating polynomial is further improved by using the Newton-Raphson method with an approximate derivative, as follows:

$$s'_{E,k} = s'_{E,k-1} - \frac{\left.\frac{d\lambda}{ds}\right|_{s'_{E,k}}}{\frac{1}{\delta s}\left(\left.\frac{d\lambda}{ds}\right|_{s'_{E,k}+\delta s} - \left.\frac{d\lambda}{ds}\right|_{s'_{E,k}}\right)} \qquad (3\text{-}31)$$

where $s'_{E,k}$ is the estimated arc length position of the local extremum after k iterations, and δs is an appropriate step for evaluation of the derivative in the denominator. (The subscript E in these equations denotes the extremum.) The interpolated value of the curve point at the extremum, evaluated after convergence of s'_E, is denoted by $\vec{u}'_E$.

Finally, the value of the curve point at the extremum is precisely calculated to within the specified tolerance by solving the equations

$$o_i^0 + \lambda\left(o_i^1 - o_i^0\right) - C_i(\vec{x}) = 0,\ i = 1, \ldots, 6$$
$$\det\left[\frac{\partial C_k}{\partial x_j}\right] = 0 \qquad (3\text{-}32)$$

At the critical point, the observation partial derivatives matrix is singular (see Equation (3-24) with $d\lambda/ds = 0$), hence the vanishing of its determinant, which is expressed by the seventh equation above. Using $\vec{u}_E'$ as an initial estimate, Equations (3-32) are solved, using standard Newton-Raphson iteration, to yield the final value of the critical point $\vec{u}_E$. (The partial derivatives of the determinant required for the Newton-Raphson method are evaluated approximately by numerical differencing.)

3.5.5 TERMINATION

At each step, this algorithm determines if the solution curve has been followed back to it starting state, which is generally a point at $\lambda = 0$, although the generalization to other starting points is straightforward.

First, the subalgorithm determines if the curve has crossed $\lambda = 0$, and, if so, the $\lambda = 0$ states are precisely calculated. This is accomplished with the subalgorithm of Section 3.5.3, with the obvious modification.

Next, each of the $\lambda = 0$ states, if any exist, is compared with the starting state, and if a match is found to within a specified precision, the algorithm terminates and indicates that the curve has been followed back to the start.

Other emergency terminations can be signalled if any of the following conditions occurs:

- The maximum allowed number of curve points is exceeded.
- The maximum allowed number of solution states is exceeded.
- The maximum allowed number of critical points is exceeded.
- The algorithm-selected step size is less than the specified minimum value.

3.5.6 STEP-SIZE SELECTION

Step-size selection is based on the idea that the amount of numerical calculation required for the correction sub algorithm (Section 3.5.8) should be roughly the same for all steps. This should make the prediction error approximately uniform resulting in small steps for sharp bends and large steps for nearly straight portions of the curve. To accomplish this, the step-size selection subalgorithm chooses the next step, $\Delta s'_{new}$, on the basis of the last step, Δs_{old}, and the number of iterations, I_{old}, that were required for the corrector to converge to the specified tolerance.

Two iteration numbers, I_{up} and I_{down}, are specified, along with two step-size adjustment factors, F_{up} and F_{down}. Typically

$$3 \leq I_{up} \leq 5$$

$$5 \leq I_{down} \leq 8$$

$$F_{up} = 1.4$$

$$F_{down} = 0.6$$

Given the values of these parameters, the next step size is determined from the following:

$$\Delta s'_{new} = F_{up}\,\Delta s_{old}, \text{ if } I_{old} \leq I_{up} \qquad \text{(increased step)}$$

$$\Delta s'_{new} = \Delta s_{old}, \text{ if } I_{up} < I_{old} < I_{down} \qquad \text{(same step)}$$

$$\Delta s'_{new} = F_{down}\,\Delta s_{old}, \text{ if } I_{down} \leq I_{old} \qquad \text{(decreased step)}$$

This selection mechanism has performed reliably in practice. However, the fact that the I's are integers sometimes leads to too coarse an adjustment capability when an attempt is being made to optimize overall efficiency. A more sophisticated subalgorithm, in which the step size is derived directly from the curvature, torsion, and polynomial parameter, N, might enable finer control.

The initial step size can be any reasonably small value, which the starter (Section 3.5.1) will automatically reduce if necessary.

The subalgorithm makes one additional modification to the step size selected with Equations (3-33). This adjustment prevents the next curve point from jumping too far across $\lambda = 0$ or $\lambda = 1$, since curve points may need to be evaluated at those values of λ (Sections 3.5.3 and 3.5.5). In the following, $\bar{\lambda}$ represents any one of these values of λ for which a curve point should be placed nearby.

If the linearly extrapolated next curve point would be located on the other side of the $\lambda = \bar{\lambda}$ hyperplane, in relation to the last computed curve point, and if the distance of the last computed curve point from the $\lambda = \bar{\lambda}$ hyperplane is greater than a specified tolerance, the step size is specified as follows:

$$\Delta s'_{new} = 1.01\,\frac{s_1 - s_2}{\lambda_1 - \lambda_2}\,(\bar{\lambda} - \lambda_1) \qquad (3\text{-}34)$$

where the subscripts 1 and 2 denote back points. This selection will place a curve point near $\bar{\lambda}$ so as to speed the calculation of the $\lambda = \bar{\lambda}$ state.

3.5.7 PREDICTION OF NEW CURVE POINT

The Lagrange polynomial is used for interpolation or extrapolation of the state $\vec{u} \equiv (\lambda, x, y, z, \dot{x}, \dot{y}, \dot{z})^T$. The arc length, s, is the independent variable. The interpolated or extrapolated state is computed from

$$\vec{u}'(s') = \sum_{i=1}^{N} L_i(s') \vec{u}(s_i) \qquad (3\text{-}35)$$

with

$$L_i(s') \equiv \frac{\prod_{\substack{j=1 \\ j\neq i}}^{N} (s' - s_j)}{\prod_{\substack{j=1 \\ j\neq i}}^{N} (s_i - s_j)}, \quad i = 1, \ldots, N$$

where $\vec{u}(s_i)$, $i = 1, \ldots, N$, are the known curve points at $s_1, \ldots, s_N$. In prediction, $s'_{new} = s_1 + \Delta s'_{new}$, where $\Delta s'_{new}$ is selected as described in Section 3.5.6.

The number of points used to define the polynomial, denoted by N, is a parameter. Through experience, values for N of 3 or 4 work best. If N is too large, then predictions become poor near sharp bends in the solution curve, leading to poor convergence and the necessity to repeat the step.

At the completion of the starter algorithm, N is set to the value 2. After completion of each additional predictor-corrector step, N is incremented by one until the specified value is reached. The algorithm for the collection of critical points requires that N be three or greater.

The vector tangent to the polynomial fit is required in the corrector subalgorithm and the monitoring subalgorithm (Sections 3.5.8 and 3.5.9). Differentiation of Equation (3-34) yields the necessary formula:

$$\left.\frac{d\vec{u}'}{ds}\right|_{s'} = \sum_{i=1}^{N} \left.\frac{dL_i}{ds}\right|_{s'} \vec{u}(s_i) \qquad (3\text{-}36)$$

with

$$\left.\frac{dL_i}{ds}\right|_{s'} = \frac{1}{\prod_{\substack{j=1 \\ j\neq i}}^{N} (s_i - s_j)} \sum_{\substack{k=1 \\ k\neq i}}^{N} \left| \prod_{\substack{j=1 \\ j\neq i \\ j\neq k}}^{N} (s' - s_j) \right|, \quad i = 1, \ldots, N$$

3.5.8 CORRECTION OF NEW CURVE POINT

Given the predicted state, u', at s', a precise solution for Equations (3-3), subject to one constraint equation, is computed with the Newton-Raphson method. The constraint requires that each correction, $\delta\vec{u} = (\delta\lambda, \delta x, \delta y, \delta z, \delta\dot{x}, \delta\dot{y}, \delta\dot{z})^T$, must lie in the five-dimensional hyperplane that is perpendicular to the tangent vector of polynomial approximation at the predicted state. (The N-point polynomial approximation for the tangent vector is based on the predicted point and N-1 backpoints.)

Using this constraint, the correction, $\delta\vec{u}$, is computed by solving the following 7x7 linear system:

$$\left.\frac{d\vec{u}}{ds}\right|_{s'} \cdot \delta\vec{u} = 0$$

$$\left(o_i^1 - o_i^0\right)\delta\lambda - \sum_{j=1}^{6} \frac{\partial C_i}{\partial x_j}\,\delta x_j = -\left[o_i^0 + \lambda\left(o_i^1 - o_i^0\right) - C_i(\vec{x})\right] \qquad (3\text{-}37)$$

$$i = 1, \ldots, 6$$

where $\delta\vec{u} \equiv (\delta\lambda, \delta\vec{x})^T$. These linear equations are solved by standard Gaussian elimination with row and column pivoting.

The tangent vector

$$\left.\frac{d\vec{u}}{ds}\right|_{s'}$$

is updated so as to be calculated from the current iterated state, rather than the original predicted state, if during the iterations the tangent vector changes by more than a specified tolerance. Thus, the perpendicularity condition is only approximate.

The iterative corrections are continued until any one of the following conditions occurs:

$$|\delta\vec{u}| < \varepsilon_1 \qquad (3\text{-}38)$$

or

$$\max_{1 \le i \le 6} \left| \frac{o_i^0 + \lambda(o_i^1 - o_i^0) - C_i(x)}{\max\left(\left|o_i^0\right|, \left|o_i^1\right|, \left|C_i\right|\right)} \right| \le \varepsilon_2 \qquad (3\text{-}39)$$

where ϵ_1 and ϵ_2 are specified parameters. Typically

$$\epsilon_1 = 10^{-13}$$

$$\epsilon_2 = 10^{-14}$$

If neither of these conditions is satisfied, and the specified maximum number of iterations (typically 6 to 8) is reached, then the subalgorithm returns a nonconvergence signal, which causes selection of a smaller step, $\Delta s'_{new}$.

3.5.9 MONITORING OF NEW CURVE POINTS

This subalgorithm monitors the computation of new curve points and directs the recomputation of a curve point with a reduced step size

$$\Delta s'_{revised} = F_{reduce} \cdot \Delta s'_{new} \qquad (3\text{-}40)$$

if one of the following conditions has occurred:

1. The corrector subalgorithm has not achieved convergence with the original step.
2. The change in the tangent vector between the new curve point and the last back point exceeds a specified tolerance.

Typically, $F_{reduce} = 0.5$.

The second condition is checked with

$$\left.\frac{d\vec{u}}{ds}\right|_{s_1} \cdot \left.\frac{d\vec{u}}{ds}\right|_{s'} < 1 - \epsilon \qquad (3\text{-}41)$$

where ϵ is typically in the range 0.01 to 0.05. A very small value for ϵ should not be selected because the error in the polynomial approximation does not approach zero as the single step size, $\Delta s'_{new}$, approaches zero. Although not done in the existing algorithm, the tangent vector du/ds would be better calculated in a future algorithm through the use of Equations (3-24) rather than the polynomial approximation. This would avoid occasional difficulties encountered with tangent vectors evaluated from the polynomial approximation.

3.5.10 CONDITIONS OF FAILURE OF THE CURVE-FOLLOWING ALGORITHM

Through considerable testing, three conditions under which the curve-following algorithm can occasionally fail have been isolated. The conditions are expected to occur in practice only very rarely, since (except for condition (3) below), they require critical adjustment of the a priori state vector to force a failure. Condition (3) occurs when the orbit is almost undetermined with the given observations, and even if this difficulty is overcome, the determined orbit will inevitably have extremely large errors.

The three conditions are as follows:

1. The a priori orbit state vector is very close to a solution state vector.
2. The a priori state vector is such that the solution curve has two loops that nearly touch, or a single loop that is nearly pinched off.
3. The six given observations are such that there are two solution states that are very close together.

Condition 1 causes the differences $O_i^1 - O_i^0$ to be very small. Assuming that most solution loops have, very roughly, the same size, when measured by the range of variation of $O_i^\lambda \equiv O_i^0 + \lambda (O_i^1 - O_i^0)$, this smallness of $O_i^1 - O_i^0$ implies

that the range of λ will be very large. Values of λ as large as 10^{12} can be reached when the a priori state vector is in error by only 10 meters. Thus, in such a case, the λ variable has not been properly scaled, and this leads, through machine precision limitations, to slow convergence of the corrector when the curve reaches large magnitudes of λ. Fortunately, the curve will be followed first through a solution state before the difficulty occurs, and this solution state is almost certainly the one desired. Proper scaling of λ could reduce this problem in the future.

Condition 2 can lead to an ambiguity concerning which of the two branches the algorithm will follow upon passing the touching point. Of course, two loops will never exactly touch, so that, in principle, the step size could be selected small enough to avoid the problem. But using very small step sizes indiscriminantly is inefficient. Better alternatives, for a very sophisticated algorithm, would be to have the algorithm check for the possible near existence of such touching points and then to either change the step size tolerances locally, when necessary, or to change the a priori state vector by an amount large enough to avoid the problem.

Condition 3 can either cause convergence difficulties in the subalgorithm for collecting solution states (Section 3.5.4) or can produce an inconsistency (only one solution state recovered for a case in which the curve doubles back). Like Condition 2, it can be corrected with sufficiently small steps, locally, in a more sophisticated subalgorithm that can reliably diagnose the condition. The condition can be diagnosed by measuring the angle between the curve tangent and the $\lambda = 1$ hyperplane.

Of the three conditions, the third is the most likely to occur in practice, when an attempt is made to determine an orbit with insufficient tracking. Even though the determined orbit may have large errors, the operational early orbit algorithm should carefully handle this case, because even relatively poor knowledge of the orbit may sometimes be good enough to lead to additional contacts and tracking of an errant spacecraft.

SECTION 4 - NUMERICAL EXAMPLES OF THE BASIC HOMOTOPY METHOD

Spacecraft tracking by means of a single TDRS (at longitude 41 degrees west) was exclusively considered. Section 4.1 provides an example with simulated tracking for a high-eccentricity orbit. Section 4.2 considers Landsat-4, using real TDRSS tracking measurements. That section includes a detailed description of one case along with less detailed results from several other test cases.

The physical modeling for the examples in Sections 4.1 and 4.2.1 was simple. Two-body orbit propagation and geometrical measurement modeling, as described in Sections 3.2.1.1 and 3.2.2.1, were used, with no corrections applied. For the simulated tracking measurements used in the example of Section 4.1, Gaussian white noise with standard deviations of 0.5 millimeters per second for Doppler and 0.5 meters for range (values typical of current TDRSS performance) was added to the simulated measurements. A 10-meter bias was added to the simulated range measurements.

Section 4.2.2 describes the accuracy improvement obtainable for the early orbit through the use of the Brouwer-Lyydane orbit propagator and the inclusion of the light propagation effects and the spacecraft transponder delays.

Typical central processing unit (CPU) times for the calculation of one complete solution loop ranged from 0.5 to 2 minutes, using the developmental program in the VAX 11/780 computer. However, in an operational implementation of the method, the CPU times would be much less because of optimization of the algorithm and because, in most cases, computation of a complete solution loop is not necessary.

4.1 ECCENTRIC ORBIT WITH SIMULATED TDRSS TRACKING

The test orbit has a perigee close to the Earth's surface and an apogee altitude of 12,000 kilometers, which is near the outer limit of complete TDRSS coverage. The orbit period is 230 minutes. Simulated Doppler observations at 0, 20, 40, 60, 80, and 100 minutes were used to generate the solution curve.

The truth model orbit elements, the a priori estimate, and the four solutions along the solution curve are listed in Table 4-1. The projection of the solution curve onto the λ - z plane is shown in Figure 4-1. As this is a two-dimensional projection of a simple curve that lies in a seven-dimensional space, the apparent cusps and self-intersections are illusory.

Although solution 1 is very close to the true state, the other three solutions also exactly fit the six given observations. Other information would be required to select the correct solution from among the four. For example, the six pairs of TDRSS antenna pointing angles should be sufficient to make the correct selection.

The near symmetry in Figure 4-1 is attributable to the symmetry inherent in TDRSS range and Doppler orbit determination. The range and Doppler measurements are unchanged under the transformation

$$\begin{Bmatrix} z', \dot{z}' \end{Bmatrix} \rightarrow \begin{Bmatrix} -z', -\dot{z} \end{Bmatrix}$$
$$\begin{Bmatrix} x', y', \dot{x}', \dot{y}' \end{Bmatrix} \rightarrow \begin{Bmatrix} x', y', \dot{x}', \dot{y}' \end{Bmatrix} \qquad (4\text{-}1)$$

where z' is the position coordinate measured along the direction that is normal to the TDRS orbit plan and x' and y' are coordinates in the TDRS orbit plane. Also, any two-body trajectory subjected to this transformation yields

Table 4-1. A Priori Estimate, True State, and Solution States for the Eccentric Test Case

Keplerian Element	A Priori Estimate	True State	Solution 1	Solution 2	Solution 3	Solution 4
a (km)	10,000	12,500	12,499.997	18,696.	12,499.997	18,696.
e	0.5	0.44	0.44000	0.608	0.44000	0.608
i (deg)	15	10	10.000	84.9	13.079	85.4
Ω (deg)	140	145	144.998	206.5	320.117	26.3
ω (deg)	-80	-90	-89.998	185.1	94.98	8.3
M (deg)	-40	-35	-35.000	345.3	-35.000	345.3

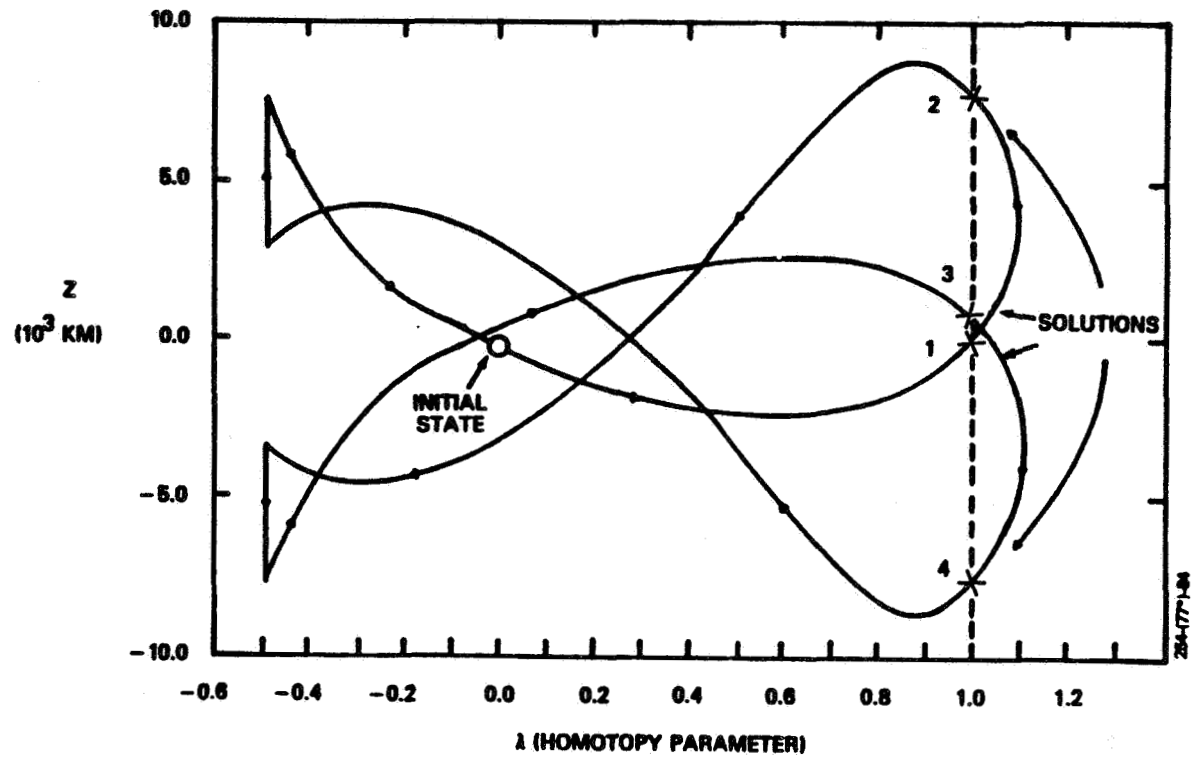


Figure 4-1. Projection of the Solution Curve Onto the λ - z Plane for the Eccentric Test Case

another, z'-reflected two-body trajectory; hence mirror symmetry in the solution curves is therefore present. (Actually, Figure 4-1 is slightly asymmetric because of the small nonzero inclination and the eccentricity of the TDRS orbit.)

The high eccentricity in this test case does not introduce any special considerations into the determination of the solution by this method. In fact, solution curves often have portions on which the orbit states are hyperbolic, even if the a priori estimate and the solution states are not. Except for details such as the shapes of the solution curves and the number of solutions, the range and Doppler problems all show the same general character: smooth, closed solution curves and an even number of solutions (except in the rare case that a solution curve tangentially touches the $\lambda = 1$ hyperplane).

In this example, the solution curve was represented numerically by a sliding quadratic polynomial fit at 125 steps. Ninety seconds of CPU time in the development program were required on the VAX 11/780 computer.

4.2 LANDSAT-4 WITH REAL TDRSS TRACKING

4.2.1 SIMPLE MEASUREMENT AND TRAJECTORY MODELING

The Landsat-4 orbit is near polar and circular, with a radius of 7070 kilometers. Only a very limited amount of TDRSS tracking was available: two or three 15-minute data groups per day with one-revolution spacing between the groups. The six measurements used in this example consisted of a single range measurement midway between two Doppler measurements in each of two consecutive data groups. These observations were selected at 0.5, 6.5, 13.0, 97.0, 102.0, and 107.5 minutes from the reference time, 14^h56^m on March 14, 1984.

The initial estimate, actual solution (osculating elements), and the four solutions found are listed in Table 4-2. Each of the four solutions was compared, over a 100-minute comparison interval, with a moderately precise Landsat-4 orbit solution calculated with the Goddard Trajectory Determination System (GTDS) Differential Correction (DC) Program. The position accuracy of this solution, which was based on NASA S-band ground tracking alone, is about 0.1 kilometers. The resulting position differences are shown in the bottom of Table 4-2. These differences are extremely large, except for Solution 2, which corresponds to the actual Landsat-4 orbit. Nearly all of the 50-kilometer error in Solution 2 is due to the two-body approximation. As indicated earlier, the correct solution of the four should be selected using the known values of the TDRS antenna beam angles.

Projections of the solution curve on the λ - y and λ - z planes are shown in Figures 4-2 and 4-3, respectively. In this example, the solution curve consists of two disjoint loops, which have the mirror symmetry discussed in Section 4.1. (The second loop has been suppressed in Figure 4-2 for clarity; it nearly coincides, in its λ - y projection, with the first loop.) Each one of the mirror-symmetric loops can be obtained from the other by using transformation (4-1). After computation of one of the loops, the existence of the other can be inferred by the absence of the expected mirror symmetry in the two solutions.

The computations for this case used 81 predictor-corrector steps for the loop and 46 seconds of VAX CPU time in the developmental program.

Results from several other Landsat-4 cases in which the tracking schedule and the a priori estimate were varied are summarized in Tables 4-3 through 4-5. All of these cases had the same reference time as the previous example and also

Table 4-2. A Priori Estimate and Solution States for Landsat-4 Test Case

Keplerian Element	A Priori Estimate	Actual Solution 1	Solution 1	Solution 2	Solution 3	Solution 4
a (km)	8000	7073.7	6960.5	7083.7	6969.5	7083.7
e	0.01	0.00097	0.819	0.00042	0.819	0.00042
i (deg)	45	98.2	79.2	98.2	76.4	100.9
Ω (deg)	0	137.4	312.1	137.5	132.2	317.6
ω (deg)	0	197.5	247.8	333.2	67.4	153.8
M (deg)	0	100.1	22.6	146.5	22.6	146.5
Maximum Error (km)	14,904		19,726	50.6	13,480	13,974

[1]From GTDS Differential Correction Solution.

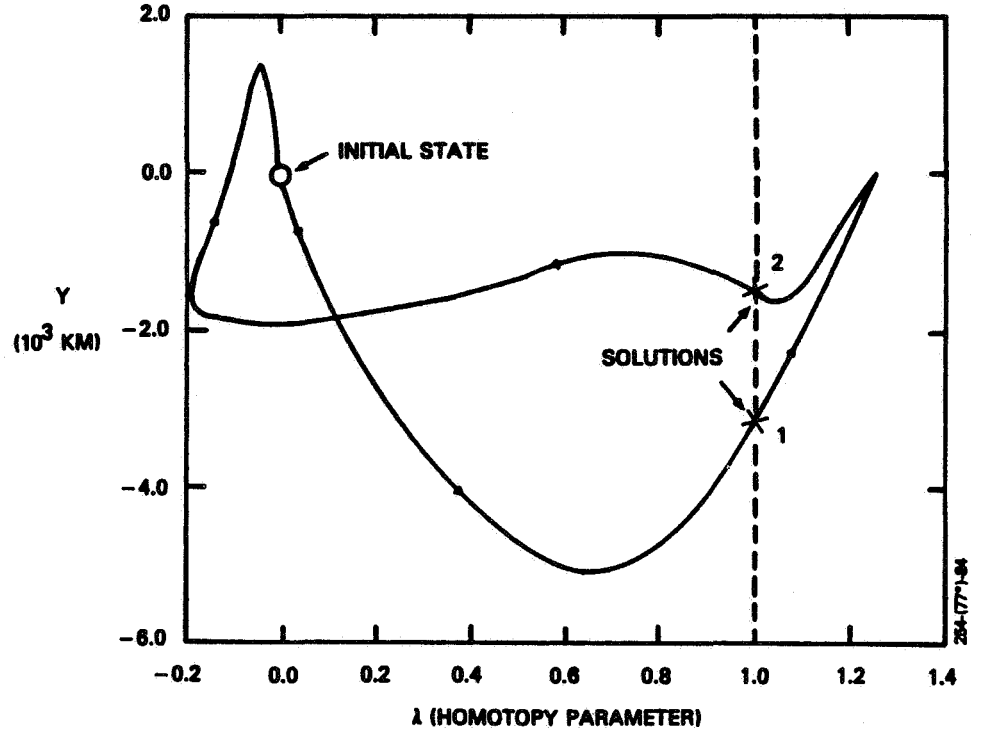


Figure 4-2. Projection of the Solution Curve Onto the λ - y Plane for the Landsat-4 Test Case

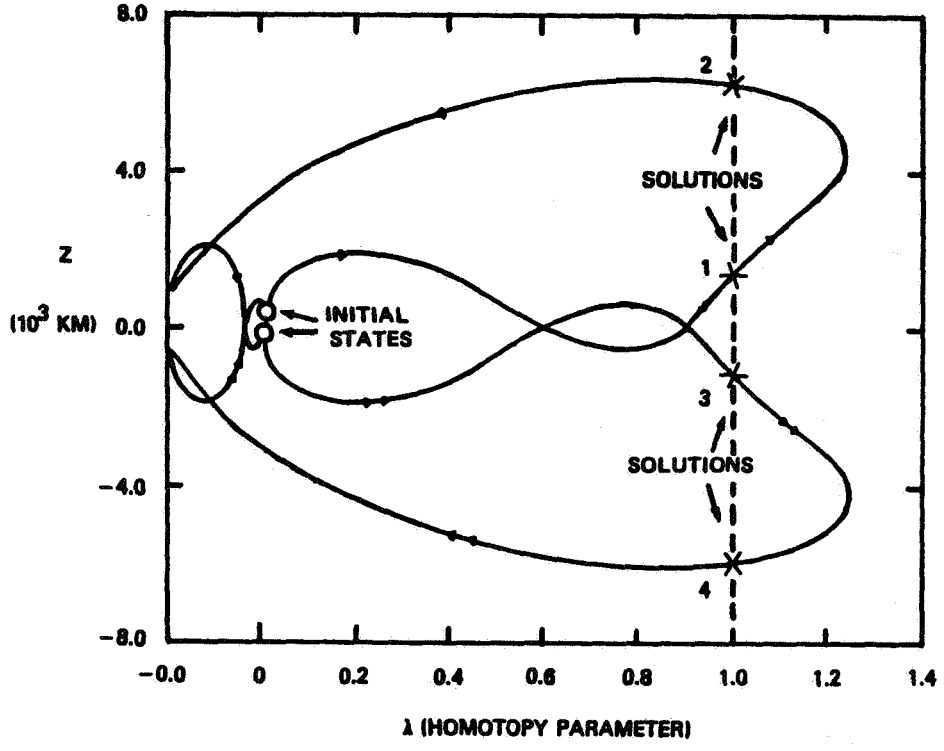


Figure 4-3. Projection for the Solution Curve Onto the λ - z Plane for the Landsat-4 Test Case

Table 4-3. A Priori Estimates Used for Landsat-4 Early Orbit Determination

Keplerian Element	A Priori Estimate A	A Priori Estimate B
a (Km)	7000	8000
e	0.01	0.01
i (deg)	95	45
Ω (deg)	135	0
ω (deg)	0	0
M (deg)	115	0

Table 4-4. Landsat-4 Early Orbit Determination Results for One-Revolution Tracking

Measurement Types[1] (1 = Range, 2 = Doppler)	A Priori Estimate[2]	Maximum Errors for All Solutions (km)[3]
1, 1, 1, 1, 1, 1	A	[46.1, 15187, 13969, 20112]
1, 1, 1, 1, 1, 1	B	[20112, 46.1] [15187, 13969]
2, 2, 2, 2, 2, 2	A	[55.9, 20299], [13972, 15670]
2, 2, 2, 2, 2, 2	B	Solutions not reached. $\lambda_{max} = 0.85$
2, 1, 2, 2, 1, 2	A	[50.9, 14018], [13976, 19863]
2, 1, 2, 2, 1, 2	B	[14018, 50.9], [19863, 13976]

[1]Types are designated in time order, from the reference time. In all cases the measurement times were 1.0, 6.5, 12.5, 97.5, 102.5, and 107.5 minutes.

[2]See Table 4-3.

[3]The brackets group solutions on the same loop, in the order found. In cases with two loops, the two solution loops are mirror images with respect to the TDRS orbit plane.

Table 4-5. Landsat-4 Early Orbit Determination Results for a Very Short Tracking Interval

Measurement Types[1] (1 = Range, 2 = Doppler)	A Priori Estimate[2]	Maximum Errors for All Solutions (km)[3]
1, 1, 1, 1, 1, 1	A	Solutions not reached. $\lambda_{max} = 0.982$
1, 1, 1, 1, 1, 1	B	Solutions not reached. $\lambda_{max} = 0.998$
2, 2, 2, 2, 2, 2	A	[1072, 3273], [14033, 14027]
2, 2, 2, 2, 2, 2	B	[3273, 1072, 14027, 14033]
1, 1, 1, 2, 2, 2	A	Solutions not reached. λ_{max} 0
1, 2, 1, 2, 1, 2	A	Solutions not reached. λ_{max} 0
1, 1, 1, 2, 2, 2,	A	[1126, 2194], [14010, 14033]

[1]Types are designated in time order, from the reference time. In all cases, the measurement times were 0.5, 3.0, 5.5, 8.0, 10.5, and 13.0 minutes.

[2]See Table 4-3.

[3]The brackets group solutions on the same loop, in the order found. In cases with two loops, the two solution loops are mirror images with respect to the TDRS orbit plane.

used the same simple trajectory and observation models. The two a priori estimates that were tried are given in Table 4-3. Estimate A is relatively close to the Landsat-4 orbit, while Estimate B is not. Table 4-4 gives the results for cases that have three measurements selected from the first data group and the second three measurements selected from the second data group, which occurs about one orbital revolution later. Table 4-5 gives the results for cases that have the six observations entirely in the first data group. The errors in the solutions that are listed in these tables are the maximum total position differences that were measured in a 100-minute comparison with the GTDS differential correction solution.

Table 4-4 confirms that the solutions obtained are independent of the a priori estimate. Second, the errors for the "actual" solution, which are about 40 to 50 kilometers, do not depend significantly on whether range or Doppler measurements are used. Most of the error is attributable to the simple trajectory model; this is shown in Section 4.2.2. For the fifth case in Table 4-4, solutions were not obtained; the solution loop did not extend beyond $\lambda = 0.85$. Presumably, the solutions lie on a different loop.

For Table 4-5, the time span of the tracking was 13 minutes. In this case, the solutions are very close to the critical points (as indicated by the pairing of the maximum error values) and, therefore, the problem is ill-conditioned. The solutions are then very sensitive to measurement and modeling errors, and, as the results show, good solutions cannot be obtained. Presumably, solutions do not even exist for the first two cases in Table 4-5. The results indicate that 13 minutes is too short a span to determine a good orbit with TDRSS range and Doppler tracking (assuming tracking from a single TDRS). However, in some cases, even a poor orbit may be sufficient to enable subsequent acquisition and collection of additional tracking.

4.2.2 IMPROVED MEASUREMENT AND TRAJECTORY MODELING

In this section, it is demonstrated that the 50-kilometer error in the one-revolution orbits determined for Landsat-4 (Section 4.2.1) is attributable primarily to the two-body approximation and omission of light travel time effects and that most of the error can be eliminated with better modeling.

To assess physical accuracy, three types of Landsat-4 ephemerides were compared. These three types are as follows:

1. The GTDS differential correction reference solution, which is the precise STDN solution described previously (0.1-kilometer accuracy).

2. Solutions computed with the developmental program using the homotopy method of this study. Options varied were two-body/Brouwer-Lyydane trajectory model and light-time/no-light-time measurement modeling. (The transponder delay correction was not performed for the testing but is easily included.)

3. GTDS differential correction solutions using the same six measurements as in type 2. The GEM9 5x0 gravity model, with the Cowell propagator, was used in GTDS to approximate the Brouwer-Lyddane. All GTDS runs included light-time modeling for the measurements. The transponder delay correction was switched on and off.

For the six-observation solutions, the reference time was set at 14^h56^m on March 14, 1984, and the six range observations were selected at 0.5, 6.5, 13.0, 97.0, 102.0, and 107.5 minutes after the reference time, the same as for the results of Section 4.2.1. In both the GTDS six-observation solutions and the homotopy method solutions, the TDRS state

was propagated over the data span with the two-body propagator. The initial estimate for the GTDS runs was sufficiently close to the solution to obtain convergence for the GTDS six-observation solutions.

Comparisons for two-body solutions are listed in Table 4-6. The first two rows of the table indicate that the light-time correction by itself is not responsible for the 50-kilometer error with range tracking only, and the differences in the second two rows indicate that the omission of the spacecraft transponder delays is also not responsible. Each of these two effects evidently contributes errors of only 1 to 2 kilometers. Finally, the last line in Table 4-6 shows that the physical modeling in GTDS and the developmental program are closely matched when two-body propagation is used. The 60-meter difference present in this last comparison is regarded as sufficiently small that it can be neglected for the purposes of early orbit determination. It is due to small, systematic differences, such as in the values of GM and the speed of light.

Comparisons for the Brouwer-Lyddane propagator are shown in Table 4-7. The ephemeris differences listed there make it clear that the previous 50-kilometer error is attributable mostly to error in the gravity model. Furthermore, as was indicated by Table 4-7, the transponder delay correction and light-time modeling each contributes 1 to 2 kilometers of error. The differences in Line 5 in Table 4-7 are not so small as the 60-meter differences in Table 4-6 because the GEM9 5x0 model, with the Cowell propagator, matches the Brouwer-Lyddane propagator only approximately; hence, the agreement is therefore not as close in this case.

It should be noted that complete solution loops generally cannot be computed using the Brouwer-Lyddane propagator because, as formulated, high-eccentricity orbits cannot be

Table 4-6. Comparisons Among Landsat-1 Orbit Solutions (Two-Body Orbit Propagation Evaluated)

Solution A	Solution B	Transponder Delay		Light-Time Correction		Maximum Position Differences[1] Between A and B (km)			
		A	B	A	B	Radial	Cross-Track	Along-Track	Total
GTDS (Ref)	Homotopy	On	Off	On	Off	21.55	17.73	44.89	47.34
GTDS (Ref)	Homotopy	On	Off	On	On	21.56	17.98	43.25	44.81
GTDS (Ref)	GTDS (6-OBS)	On	Off	On	On	21.55	18.00	43.28	45.84
GTDS (Ref)	GTDS (6-OBS)	On	On	On	On	21.77	17.30	41.73	44.42
GTDS (6-OBS)	Homotopy	Off	Off	On	On	0.011	0.021	0.056	0.060

[1]Ephemeris comparisons were 100 minutes long.

Table 4-7. Comparisons Among Landsat-1 Orbit Solutions (Brouwer-Dyddane Orbit Propagation Evaluated)

		Transponder Delay		Light-Time Correction		Maximum Position Differences[1] Between A and B (km)			
Solution A	Solution B	A	B	A	B	Radial	Cross-Track	Along-Track	Total
GTDS (Ref)	Homotopy	On	Off	On	Off	0.738	0.405	2.509	2.525
GTDS (Ref)	Homotopy	On	Off	On	On	0.713	0.673	1.834	1.845
Homotopy	Homotopy	Off	Off	On	Off	0.094	0.311	1.862	1.864
GTDS (Ref)	GTDS (6-OBS)[1]	On	Off	On	On	0.722	0.593	2.111	2.123
Homotopy	GTDS (6-OBS)[1]	Off	Off	On	On	0.060	0.137	0.287	0.308
GTDS (6-OBS)1	GTDS (6-OBS)[1]	On	Off	On	On	0.463	0.847	2.310	2.443

[1]The GEM9 gravity model, truncated to 5x0, was used to approximate the Brouwer-Lyddane propagator (applied to target satellite only) in GTDS.

[2]Ephemeris comparisons were 100 minutes in length.

handled. Thus, the two-body solutions must be computed first; the Brouwer-Lyddane model is then used to refine the selected solution by computing only the portion of the solution loop from $\lambda = 0$ to $\lambda = 1$. Alternatively, ordinary Newton-Raphson iterations would also work for refinement.

SECTION 5 - PROTOTYPE ALGORITHM FOR TDRSS EARLY ORBIT DETERMINATION

This section considers construction of a fairly complete, automatic early orbit algorithm, based on the homotopy method for solving the equations. The algorithm is described in Section 5.1; test results are presented in Section 5.2; and, finally, the limitations of the algorithm and suggestions for improvement are presented in Section 5.3.

5.1 ALGORITHM DESCRIPTION

The algorithm requires two range measurements, four additional range or Doppler measurements, and two pairs of antenna beam angles that coincide in time with the two range measurements. The two range-azimuth-elevation triples are used to derive a very rough state vector if a rough a priori is not available. This rough state vector is subsequently used to initiate an attempt to determine a more precise orbit that utilizes the range and/or Doppler data. Resultant orbit solutions are screened using the angle data to eliminate extraneous solutions. Acceptable solutions are refined by computing a partial solution curve, using improved trajectory and observation models. Thus, the algorithm has three stages.

The algorithm consists of the following steps:

Step 1. Select the required tracking measurements.

Step 2. (Optional) Attempt to determine the orbit by using externally supplied a priori estimate(s) of the state vector, as follows:

2.1 Compute solution loop(s) using the a priori state vector estimate(s).

2.2 Screen solutions found on the loop.

2.3 Refine an accepted solution, if found, using the Brouwer-Lyddane propagator and observation light-time modeling. If an acceptable solution is found, exit the algorithm.

Step 3. Determine the orbit by using a priori state vector derived from the two range-azimuth-altitude measurement triples. Corrections ΔA, ΔE are attempted, in turn, until acceptable solutions are found. This step proceeds as follows:

3.1 Select the trial values of the antenna beam angle errors ΔA, ΔE.

3.2 Compute a partial solution curve from $\lambda = 0$ to $\lambda = 1$ using the two triples (ρ, A + ΔA, E + ΔE). The initial state at $\lambda = 0$ can be any reasonable orbit state.

3.3 Using the $\lambda = 1$ state from step 3.2 as the a priori estimate and using the six range or Doppler measurements, compute a solution loop.

3.4 Screen solutions found on the loop in step 3.3.

3.5 Refine an acceptable solution, if found, using the Brouwer-Lyddane propagator and observation light-time modeling. When an acceptable solution is found and refined, exit the algorithm; otherwise, go back to step 3.1 and repeat the procedure.

In step 1, an automatic procedure for selecting the observations is intended. This procedure should contain several rules of thumb for making the optimum selection. For example, near-integral multiples of one-half an orbit revolution must be avoided for the spacing between the two range-azimuth-altitude triples. In the testing that is reported in Section 5.2, this step was omitted, and the

observations were selected manually. Optionally, the automatic selection procedure can include simple tests for data validity, such as polynomial fitting and checking of residuals.

In step 2, an attempt is made to find the solution by using available estimates of the solution. More than one such estimate can be tried. In practice, it is expected that this step will nearly always yield the desired solution, (because good estimates will be available), and this step is therefore omitted from the testing in Section 5.2.

Step 3, performed if step 2 fails or is bypassed, attempts to get the orbit entirely from the tracking data, not using any externally supplied a priori estimate. This step is based on the fact that the actual angular position of the spacecraft must lie within a cone of known angular radius (0.86 degree for the SSA mode), centered upon the recorded values of the antenna beam angles. Therefore, by systematically trying various possible values of the angle errors (a pair of angle errors at each of the two observation times), an a priori state will eventually be found that is sufficiently close to the true solution so that both lie on the same solution loop in step 3.3. Thus, with enough trials, the solution, if it exists, will eventually be found. It is this search procedure that is primarily considered in Section 5.2.

One possible grid of trial values of the beam angle errors is schematically shown in Figure 5-6. The beam radius, BMLIM, and initial grid spacing, BDEL, are parameters, and the same values for these parameters are used at each of the two observation times. The first attempt to get the solution fixes the trial error at the times of both triples to the value labelled by '1' in Figure 5-1. The search continues by trying, in sequence, the various errors at the time of the second triple, leaving the error at the first

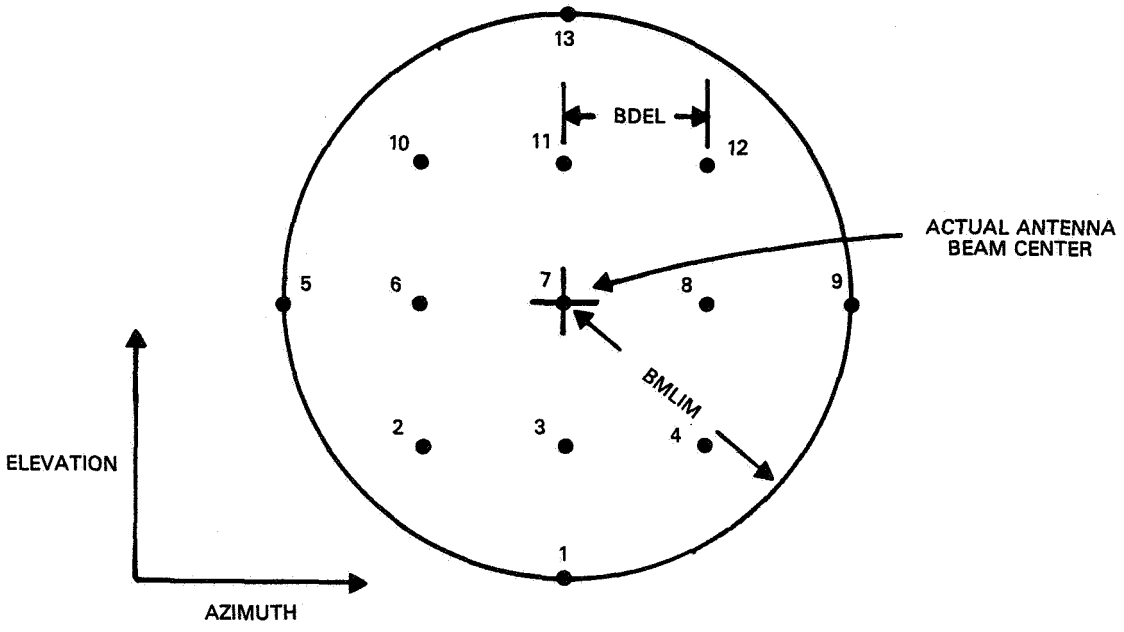


Figure 5-1. Grid Pattern for Searching for Orbit Solutions in the TDRS Antenna Beam Pattern

triple fixed. After the entire grid has been tried unsuccessfully at the second triple, the error at the first is incremented to '2', and looping is again performed at the second triple. This process is extended until all possible combinations of errors have been tried. If still no solution has been found, BDEL is reduced by 50 percent and the procedure is repeated, omitting previously tried grid points. It is not expected that the BDEL reduction procedure will ever be used in practice, but it was available for testing.

The solution screening procedure for the range/Doppler solutions checks the recorded values of the two angle pairs against those predicted by each particular orbit solution. If agreement to within a specified tolerance is found for all four angles, the solution is accepted. Because of the known TDRS orbit-plane symmetry in the set of orbit solutions (Section 4.1), the symmetric solutions are automatically generated by the algorithm and considered for screening, along with each solution actually computed. Each range/Doppler solution loop in either Step 2.1 or Step 3.3 is entirely computed before the solutions are passed, in a group, to the screening procedure. However, screening of each solution immediately after generation would speed up an operational version of the algorithm.

5.2 TEST RESULTS

5.2.1 TEST PROCEDURE

Four cases were included in Monte Carlo testing of the search part of the algorithm of Section 5.1. These are as follows:

1. Low-Inclination, Circular Orbit
2. High-Inclination, Circular Orbit
3. Low-Inclination, Elliptical Orbit
4. Landsat-4 (High Inclination, Circular)

In the first three cases, simulated observations were used, while in the fourth case, actual TDRSS tracking data was used. The orbit elements for these cases and the schedule of selected observations are given in Tables 5-1 and 5-2. For the simulated cases, a number of subcases were considered: four for each of the circular cases and twelve for the elliptical case. In these subcases, the orientation of the orbit with respect to the TDRS was varied, for example, orbit plane edge-on or face-on to the TDRS, tracking at extreme azimuth or zero azimuth, and tracking near perigee or apogee. The results for these subcases are independently tabulated in Section 5.2.2.

For each subcase, a group of 20 Monte Carlo trials was performed; a different set of azimuth and altitude errors were added to the correct antenna beam angle values in each trial. These 20 sets of errors were derived from a random number generator that simulated a rectangular probability distribution between -1.0 degree and 1.0 degree. For the trials in each of the three simulated cases, the 20 sets of applied angle errors were identical. These 20 sets of angle errors are shown in Figure 5-2. Here, the arrow points from the error at 0 minutes to the error at 25 minutes. For the Landsat-4 case, a similar set of 20 errors was added to the actual antenna beam angle values. Those are similarly shown in Figure 5-3, in which the arrow points from the error at 13 minutes to the error at 97 minutes.

The parameters specified for the algorithm to be used in the antenna beam search grid were specified as follows:

BMLIM = 1.0 degree

BDEL = 0.5 degree

The angular tolerance used for screening the solutions was specified as 1.2 degrees for both azimuth and altitude.

Table 5-1. Orbit Elements for the Four Test Cases

Keplerian Element	Low-Inclination, Circular	High-Inclination, Circular	Low-Inclination Elliptical	Landsat-4[1]
a (km)	7000	7000	12,500	7080
e	0.001	0.001	0.44	0.0006
i (deg)	10	100	10	98.2
Ω (deg)	varied	varied	varied	137.4
ω (deg)	0	0	varied	189.9
M (deg)	varied	varied	varied	281.0

[1] Reference Date: 14^h56^m on March 14, 1984.

Table 5-2. Tracking Schedules for the Four Test Cases

Test Case	Times of Selected Observations (Minutes)		
	Relay Range	Relay Doppler	Azimuth, Elevation
Low Inclination, Circular[1]	0 25 95	0 25 95	0 25
High Inclination, Circular[1]	0 25 95	0 25	0 25
Low Inclination, Elliptical[1]	0 25 95	0 25 95	0 25
Landsat-4	0.5 6.5 13 97 102 107.5	-	13 97

[1] Gaussian white noise added: σ(range) = 0.005 km, σ(Doppler) = 0.5 x 10^{-6} km/sec
Range bias added: 0.01 km

C-2

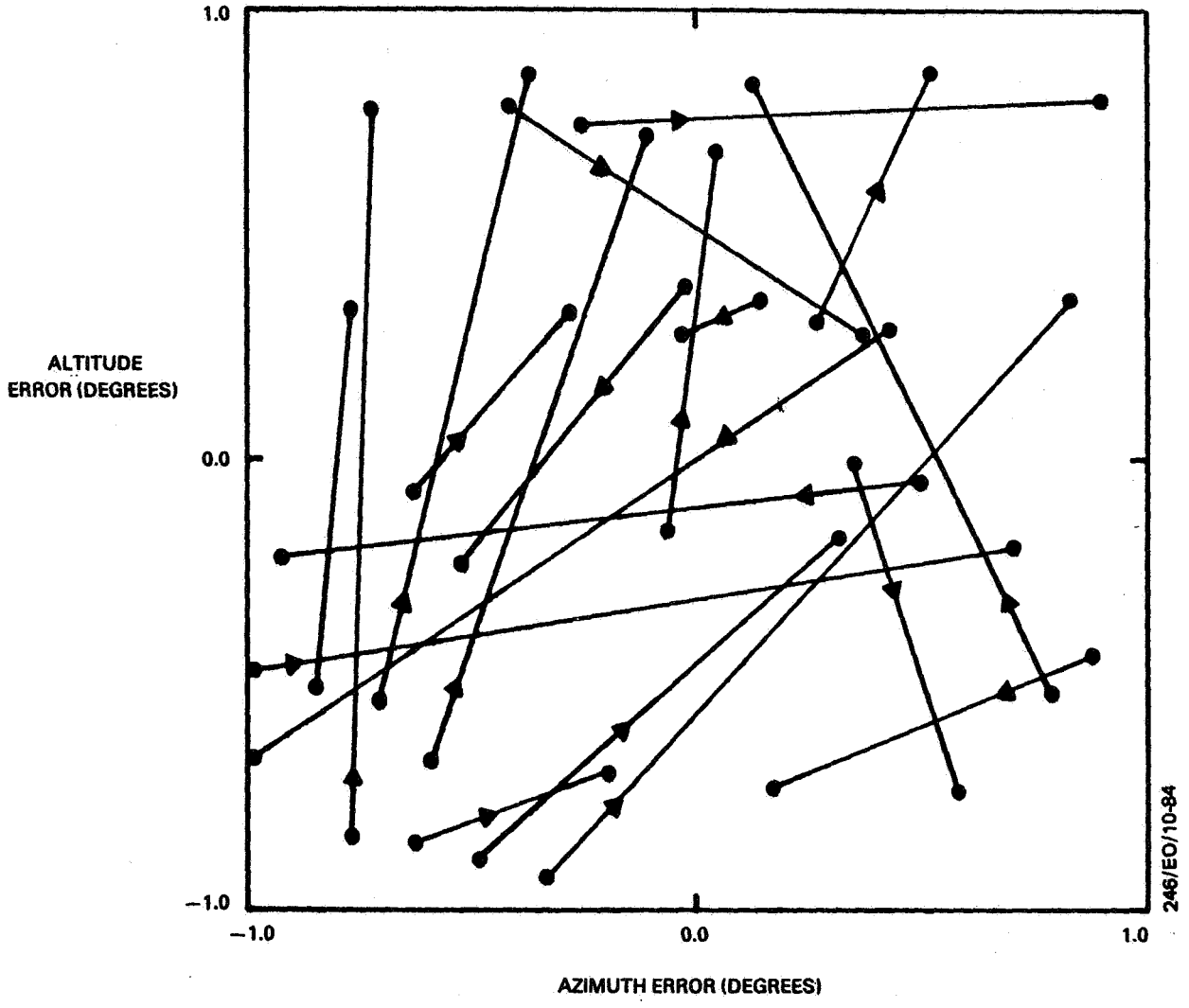


Figure 5-2. Azimuth and Elevation Error Included in the 20 Monte Carlo Trials for the Three Simulated Test Cases

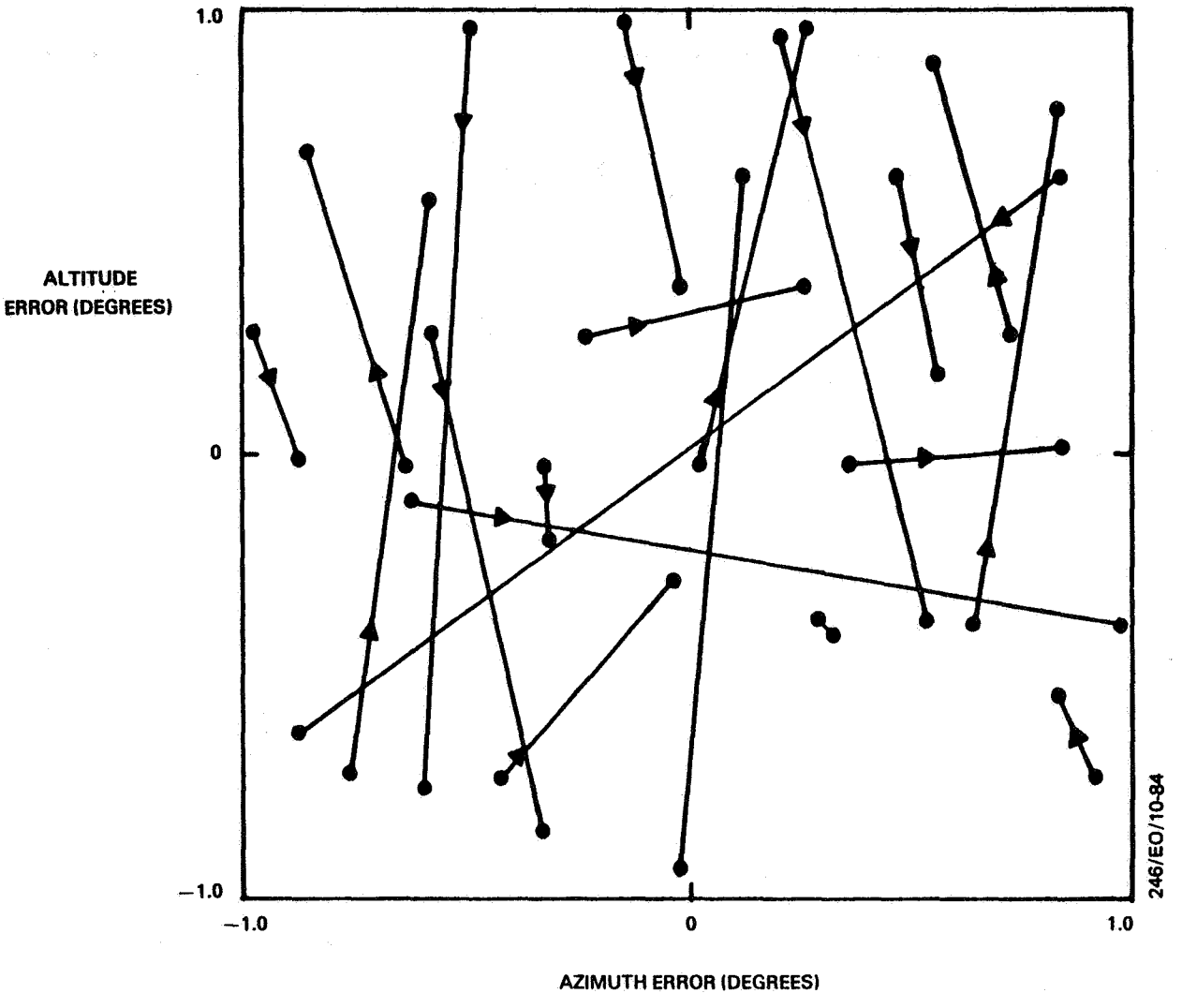


Figure 5-3. Azimuth and Elevation Errors Included in the 20 Monte Carlo Trials for the Landsat-4 Test Case

The a priori elements used in the range and angles step of the algorithm was the same for all trials and is listed in Table 5-3. The Brouwer-Lyddane refinement step was omitted in the testing.

5.2.2 RESULTS

The Monte Carlo results for the four cases are summarized in Tables 5-4 through 5-7. Each row in a table describes the outcome of the 20 trials for each subcase. For example, in the first subcase in Table 5-4, none of the trials failed to produce acceptable solutions (third column), each of the 20 trials yielded the correct solution among the acceptable solutions (Column 4), 4 trials yielded one or more extraneous solutions (but acceptable to the screening procedure) (Column 5), and the number of distinct solutions found among all of the trials was 2 (Column 6). In the last column, the number of solution loops that had to be calculated is indicated. In the first subcase, 19 trials required only one solution loop, while 1 trial required calculation of two solution loops before an acceptable solution was located.

On the whole, the search procedure was successful in locating the correct solution in all but 4 of the 420 trials. In these four trials, other extraneous (but accepted) solutions were found first, terminating the algorithm. Presumably, had the algorithm continued with the calculation of additional solution loops, the correct solution also would have been found.

In approximately one-half of the trials, the solution loop containing the correct solution also contained an accepted extraneous solution. This, along with the four failures, indicates that a tighter solution screening procedure is required.

Table 5-3. A Priori Orbit Elements Used for the Range and Angles Orbit Determination Step

Keplerian Element	A Priori Orbit Elements
a (km)	8000
e	0.001
i (deg)	45
Ω (deg)	0
ω (deg)	0
M (deg)	180

Table 5-4. Test Results for the Low-Inclination, Circular Orbit

Orbit Configuration Relative to TDRS at T = 25 Minutes		Number of Cases With No Solutions	Number of Cases With Correct Solution	Number of Cases With Other Solutions	Number of Distinct Solutions Found	Number of Loops Computed (Trials)
Orbit Plane	Azimuth					
FACE-ON	0°	0	20	4	2	1(19), 2(1)
FACE-ON	MAX	0	20	3	2	1(20)
EDGE-ON	0°	0	20	20	2	1(19), 2(1)
EDGE-ON	MAX	0	20	19	4	1(19), 2(1)

Table 5-5. Test Results for the High-Inclination, Circular Orbit

Orbit Configuration Relative to TDRS at T = 25 Minutes		Number of Cases With No Solutions	Number of Cases With Correct Solution	Number of Cases With Other Solutions	Number of Distinct Solutions Found	Number of Loops Computed (Trials)
Orbit Plane	Azimuth					
FACE-ON	0°	0	20	5	2	1(20)
FACE-ON	MAX	0	20	0	1	1(20)
EDGE-ON	0°	0	20	8	2	1(4), 2(9), 5(3) 9(1), 10(1), 13(1) 57(1)
EDGE-ON	MAX	0	20	2	2	1(14), 2(5), 5(1)

Table 5-6. Test Results for the Low-Inclination, Elliptical Orbit

Orbit Configuration Relative to TDRS T = 25 Minutes		Position in Orbit	Number of Cases	Number of Cases	Number of Cases	Number of Distinct	Number of Loops
Orbit Plane	Azimuth	At T=25 Min	With No Solutions	With Correct Solution	With Other Solutions	Solutions Found	Computed (Trials)
FACE-ON	0°	PERIGEE	0	20	11	2	1(19), 2(1)
FACE-ON	0°	IN-BETWEEN	0	20	8	2	1(17), 2(1), 4(1), 5(1)
FACE-ON	0°	APOGEE	0	20	7	2	1(19), 5(1)
FACE-ON	MAX	PERIGEE	0	20	8	2	1(20)
FACE-ON	MAX	IN-BETWEEN	0	20	7	2	1(20)
FACE-ON	MAX	APOGEE	0	19	7	3	1(15), 2(1) 4(2), 5(1)
EDGE-ON	0°	PERIGEE	0	19	18	4	1(14), 2(3), 4(1) 17(1), 53(1)
EDGE-ON	0°	IN-BETWEEN	0	18	13	4	1(15), 3(1), 4(4)
EDGE-ON	0°	APOGEE	0	20	13	2	1(20)
EDGE-ON	MAX	PERIGEE	0	20	5	2	1(20)
EDGE-ON	MAX	IN-BETWEEN	0	20	14	2	1(20)
EDGE-ON	MAX	APOGEE	0	20	17	4	1(17), 2(3)

Table 5-7. Test Results for Landsat-4

Orbit Configuration Relative to TDRS at T = 13 Minutes		Number of Cases With No Solutions	Number of Cases With Correct Solution	Number of Cases With Other Solutions	Number of Distinct Solutions Found	Number of Loops Computed (Trials)
Orbit Plane	Azimuth					
FACE-ON	MAX	0	20	8	2	1(20)

In two of the trials in Table 5-6, more than 50 solution loops and about 50 minutes of VAX CPU time were required before an accepted solution was found. This much computation is not feasible in an early orbit algorithm for operational orbit determination, and a modification of the algorithm is required to correct this. The problem can probably be corrected by using a more irregular numbering of the grid points in Figure 5-1, so that sequen- tial trials are not close together in angular position.

5.3 SUGGESTED IMPROVEMENTS AND COMMENTS

Consideration of the test results leads to three suggested improvements for the early orbit algorithm. These are as follows:

1. Tighter screening of the solutions is required. To do this, it is necessary to use more than the two angle pairs, and additional range and Doppler observations should probably be used also. Tighter screening would eliminate the four trial failures. The use of nearly all of the available range/Doppler tracking should be considered to compute a RMS weighted residual for screening. Also, additional screening for orbital energy, maximum vehicle ΔV, maximum orbital plane change, etc., should probably also be included.

2. The order of the applied beam angle errors used in the search procedures should be changed. Rather than the simple order indicated in Figure 5-1, the order should be varied so that significantly different a priori estimates will be generated from one attempt to the next. It is expected that varying the order would at least partially remedy the two trials that required more than 50 solution loops. Currently, it does not appear feasible to design a "smart" search procedure (for example, first choosing angle errors that significantly vary the mean anomaly), because such an approach turns out to be very complicated (from the

point of view of working out the equations) to implement in such a way that all possibilities are systematically treated.

3. For improvement in efficiency, the screening procedure should be applied as each candidate solution is generated. In the algorithm tested, screening was deferred until all solutions on a loop were collected. However, in most cases, the correct solution is the first one encountered along the solution loop. Therefore, immediate screening could significantly improve the efficiency of the algorithm by eliminating calculation of the remainder of the solution loop.

This algorithm does not address two of the items listed in Section 2, namely, early orbit determination for the Space Shuttle, for which range tracking is not available, and treatment of the ambiguous range observations.

Unfortunately, an analogous search procedure using two Doppler-azimuth-elevation triples will not work, because orbit solutions may not exist for any arbitrarily selected pair of such triples. This means that the necessary preliminary solution for each point in the search pattern cannot easily be obtained, as was the case for two range-azimuth-elevation triples. Therefore, at the current time, Space Shuttle early orbit determination would require externally supplied a priori estimates for computing Doppler-only solution loops.

For the problem of the redetermination of range ambiguity numbers, this study included attempts to derive equations of consistency, so that the ambiguity numbers could be derived directly from the tracking data using numerical time derivatives of the Doppler data. These attempts were unsuccessful, so that, at present, the only way to get the ambiguity numbers is with the standard algorithm, which requires an orbit state vector. In an operational program, it might

prove useful to allow overrides to increment or decrement these numbers in a manually controlled search. Alternatively, a simple software default strategy could be provided to do a limited ambiguity number search, to avoid a person in the loop.

Automatic looping over various ambiguity numbers must be used with caution in an operational program because failure of the algorithm to get a solution may be due to factors other than incorrect ambiguity numbers, namely, bad observations or insufficient tracking.

SECTION 6 - PRELIMINARY STUDIES OF EXTENSIONS OF THE BASIC HOMOTOPY METHOD

This section considers two generalizations of the method expressed by Equations (3-3). Section 6.1 contains the weighted least-squares formulation corresponding to Equations 3.3, a simple example, and a general description of test results. Section 6.2 considers a generalization of the basic homotopy method to include additional paths leading from one disjoint solution loop to another. This generalization is expected to permit the determination of the orbit solutions beginning with any arbitrary a priori estimate.

6.1 LEAST-SQUARES ORBIT DETERMINATION

6.1.1 FORMULATION

The weighted least-squares method is applied, assuming that the M measurement errors, having assigned standard deviations, $\sigma_1, \sigma_2, \ldots, \sigma_M$, are completely uncorrelated. The possible Bayesian term is not included here, but is straightforward. With these assumptions, the solution states x^1 satisfy the equations

$$\sum_{j=1}^{M} \left(\frac{1}{\sigma_j}\right)^2 \left[O_j^1 - C_j(\vec{x})\right] \left.\frac{\partial C_j}{\partial x_i}\right|_{\vec{x}} = 0, \quad i = 1, \ldots, 6 \qquad (6\text{-}1)$$

while the a priori estimate, by definition, satisfies

$$O_j - C_j(\vec{x}) = 0, \quad j = 1, \ldots, M$$

and, therefore, also satisfies

$$\sum_{j=1}^{M} \left(\frac{1}{\sigma_j}\right)^2 \left[o_j^0 - C_j(\vec{x})\right] \left.\frac{\partial C_j}{\partial x_i}\right|_{\vec{x}} = 0, \quad i = 1, \ldots, 6 \qquad (6\text{-}2)$$

The introduction of the homotopy parameter λ in the same way as in the six-observation formulation (Section 3.1) leads to the following equations, which correspond to Equations (3-3):

$$\sum_{j=1}^{M} \left(\frac{1}{\sigma_j}\right)^2 \left[o_j^0 + \lambda\left(o_j^1 - o_j^0\right) - C_j(\vec{x})\right] \left.\frac{\partial C_j}{\partial x_i}\right|_{\vec{x}} = 0, \qquad (6\text{-}3)$$

$$i = 1, \ldots, 6$$

A least-squares formulation has two main benefits for early orbit determination:

- The effects of high-frequency measurement and model errors are averaged out in a better way than in six-observation orbit determination.
- The possibility of an addition of a Bayesian term allows inclusion of a priori knowledge or constraints for the orbit. This capability can be useful when very little tracking is available.

6.1.2 SIMPLE EXAMPLE

Using Equation (6-1) in the case of one observation in the example of Section 3.3, the solution curve corresponding to the one specified by Equation (3-20) is

$$\left\{\left[1-\left(\frac{x^0}{a}\right)^2\right]^{1/2}+\lambda\left\{\left[1-\left(\frac{x^1}{a}\right)^2\right]^{1/2}-\left[1-\left(\frac{x^0}{a}\right)^2\right]^{1/2}\right\}\right\}\cdot\left\{\frac{\frac{-}{(+)}\frac{x}{a^2}}{\left[1-\left(\frac{x}{a}\right)^2\right]^{1/2}}\right\}=0 \qquad (6\text{-}4)$$

$$\left[\frac{-}{+}\left(1-\frac{x}{a}\right)^2\right]^{1/2}$$

This equation is satisfied when either one or both of the main factors is equal to zero. As in Section 3.3., the first factor is zero on the ellipse given by Equation (3-20). The second factor is zero on the straight line $x = 0$. Thus, the solution curve has two intersecting components, as shown by Figure 6-1. The two components intersect at two points, called bifurcation points. At these two points only, both factors in Equation (6-4) are equal to zero.

The existence of these bifurcation points will cause a difficulty in the curve-following algorithm of Section 3.5. At the two such points in Figure 6-1, the 2x2 matrix that corresponds to the 7x7 matrix on the lefthand side of Equation (3-36) becomes singular, and the Newton-Raphson corrector does not converge quickly near these points. Thus, the current numerical algorithm cannot follow a solution curve through a bifurcation point. A more sophisticated algorithm is required. One approach might be to patch in an analytic solution in a small region that includes the bifurcation point, introducing three new branches, each to be subsequently followed.

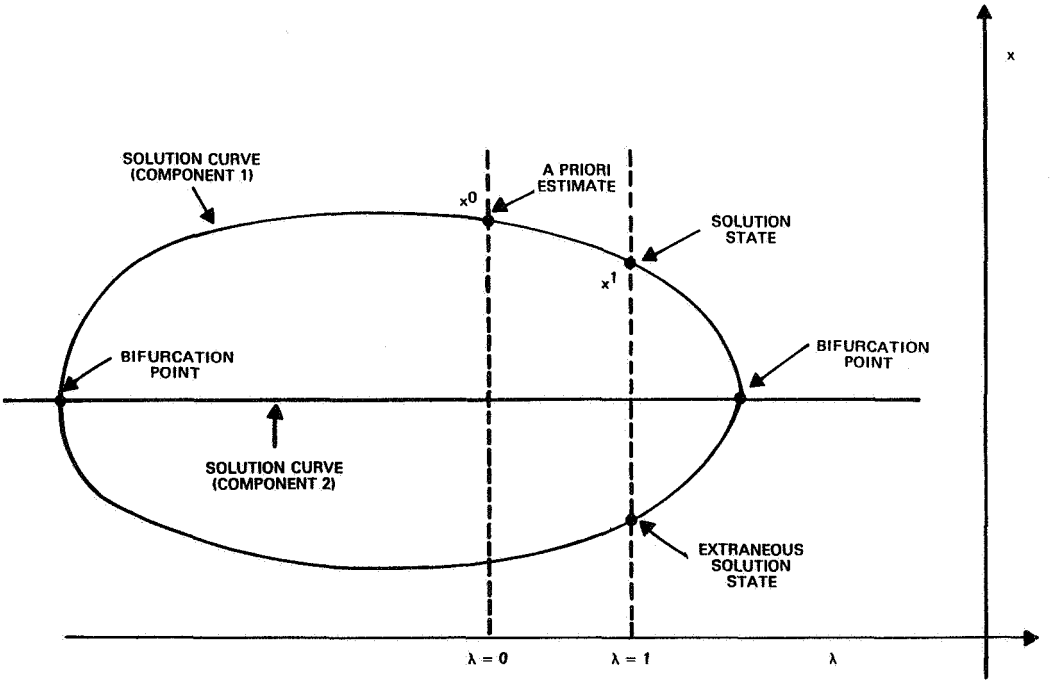


Figure 6-1. Solution Curve for the One-Dimensional Example of the Homotopy Method Using the Least-Squares Formulation

Reference 8 indicates that a second approach, using a local perturbation to remove the bifurcation point may be workable, but this has yet to be tested in the orbit determination problem.

6.1.3 DESCRIPTION OF TEST RESULTS

The picture outlined above is consistent with the results of testing in simulated TDRSS orbit determination. With pure range and/or Doppler tracking, it was found that a complete solution loop could not be followed in any case; at some point along the curve, the matrix for the Newton-Raphson corrector became singular, and the curve could not be followed beyond this point. Such points always occurred at extrema in λ. (The curve-following algorithm responds to the singularity by taking smaller and smaller steps, which soon forces termination.) In some cases, a bifurcation point was encountered before any solution state was reached. In other cases, one or more solution states were reached before a bifurcation point.

Because of the complexity introduced by the existence of the bifurcation points, further work on the least-squares method was discontinued; such additional study would be more appropriate after a thorough understanding of the six-observation case has been achieved.

6.2 METHOD FOR SYSTEMATIC CALCULATION OF DISJOINT SOLUTION LOOPS

In this section, only pure range and/or Doppler orbit determination is considered. As indicated previously, when the a priori estimate is far from the desired solution state, the two states may lie on disjoint components of the solution curve. Because of this, the basic algorithm of Section 3 may not succeed. In this section, a generalization of the formulation, which overcomes the nonconnectedness property of the loops, is described. This formulation

produces a solution curve that consists of a network of loops, in a 12-dimensional $(\lambda_1\text{-}\lambda_2\ldots\text{-}\lambda_6\text{-}\vec{x})$-space, among them the two disjoint loops containing the a priori estimate and the desired solution state. Loops of various "levels" in this network lie in various projections to lower dimension subspaces of the $(\lambda_1\text{-}\lambda_2\ldots\text{-}\lambda_6\text{-}x)$-space. Loops of neighboring levels intersect and form a connected network of loops that may be individually followed in turn, using a straightforward generalization of the numerical method of Section 3.5. This technique enables the solution state to be reached, beginning with any a priori estimate (though at an increased computational cost for poor estimates).

The basis for the generalization, the five-dimensional subspace of six-dimensional orbit state space that is here called the "critical hypersurface," is described in Section 6.2.1. Simple, numerical experiments verifying that connections between disjoint solution loops can be accomplished by paths consisting exclusively of states lying on the critical hypersurface are indicated in Section 6.2.2. Results from a two-level algorithm are summarized in Section 6.2.3. In a two-level algorithm, two distinct types of solution loops are calculated: (1) ordinary loops in $(\lambda_1\text{-}\vec{x})$ space, as defined in Equations (3-3); and (2) loops in a $(\lambda_1\text{-}\lambda_2\text{-}x)$ space, which consist (in the x-component) only of states on the critical hypersurface that satisfy Equations (3-3). For these latter loops, the image of the loop in six-dimensional observation space lies in a plane rather than on a straight line. Although it was not coded and tested in this study, the full generalization to the six-level algorithm is described in Section 6.2.4 for future reference.

6.2.1 THE CRITICAL HYPERSURFACE AND CRITICAL POINTS

Generally, six given range and/or Doppler measurements determine the six components of the orbit state. Considering six-dimensional observation (measurement) space, which has six coordinate axes representing the values of the six measurements in a fixed tracking schedule, it is readily apparent that points exist in the observation space that cannot be produced by any orbit state. The five-dimensional boundary between the regions of possible and impossible observation sets is the image of a five-dimensional hypersurface in x-space. Here, this hypersurface will be called the critical hypersurface (see Figure 6-2). It will be assumed that this surface is smooth almost everywhere, and that it is connected for nearly all range and/or Doppler orbit determination problems. This assumption remains to be proven in future study.

As suggested by Figure 6-2, a given observation set can be realized by more than one point in orbit state space. Thus, at the boundary between possible and impossible observation sets, the "possible" region "folds over" onto itself, so that "layers" are superimposed. (Of course, these "layers" are six dimensional; the six-dimensional generalization of Figure 6-2 cannot be visualized.) As the boundary of the "possible" region is reached, the number of orbit states that can realize the observation set is reduced as multiple states coalesce at the critical hypersurface. On the critical hypersurface, the six-observation orbit determination problem is ill-conditioned, that is, the observation partials matrix is singular

$$\det \left. \frac{\partial C_i}{\partial x_j} \right|_{\vec{x} \text{ on critical hypersurface}} = 0 \qquad (6\text{-}5)$$

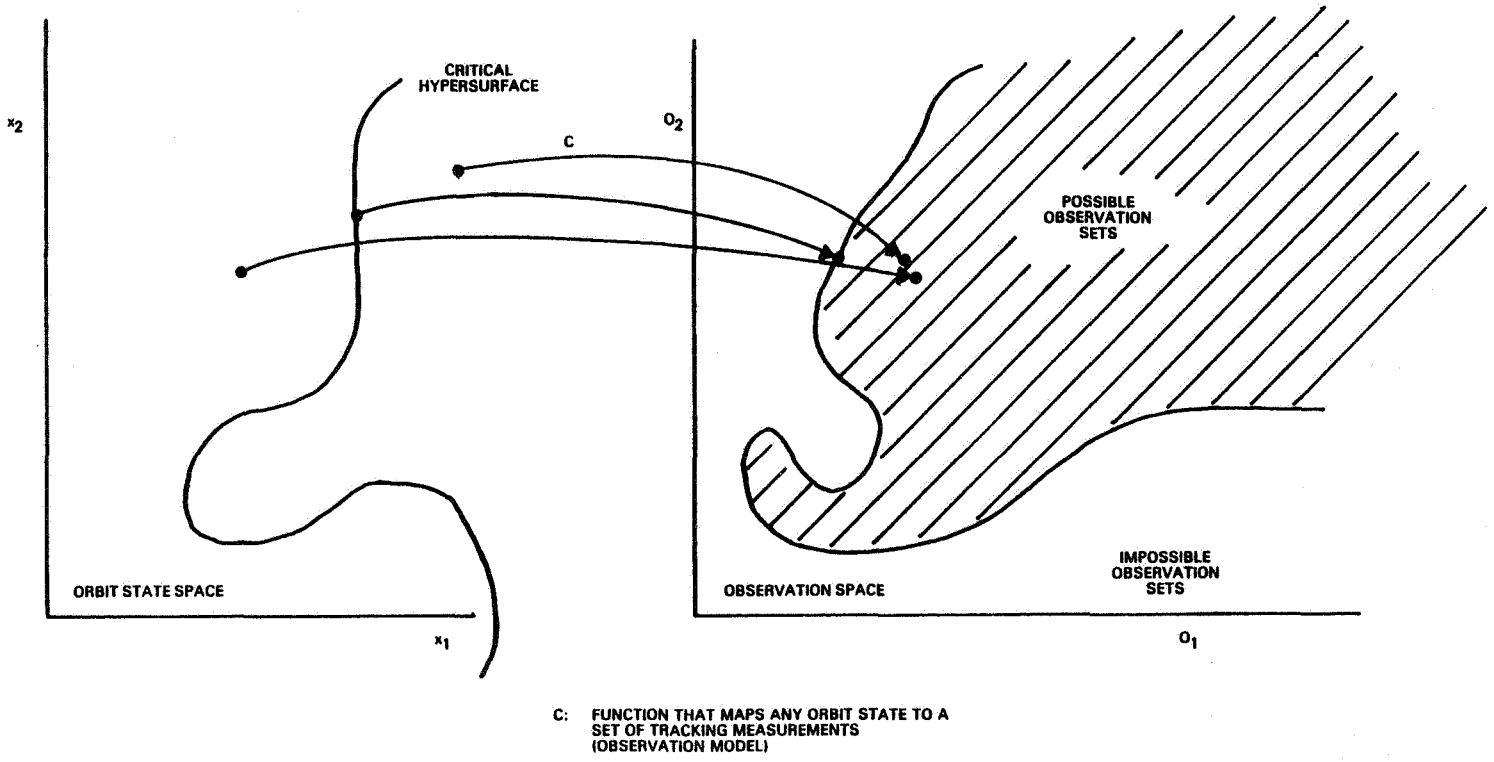


Figure 6-2. Schematic Diagram Illustrating the Critical Hypersurface

Equivalently, the six vectors $\partial C_i / \partial \vec{x}$, $i = 1, \ldots, 6$ become linearly dependent.

Equation (6-5) provides one constraint in x-space, leading to a five-dimensional hypersurface. Equation (6-5) specifies that the observation partials matrix should have rank 5 or less on this hypersurface. Lower-dimensional subspaces can be further defined by requiring that the observation partials matrix should have rank 4, rank 3, etc. In the observation space picture, this means that there is a "folded fold," a "folded, folded fold," etc., each additional folding leading to a subspace of dimension that is smaller by 1.

Next, critical points on solution loops are defined. Equation (3-3) specifies that the solution curve image in observation space should trace out a straight line. Points where this line intersects the region between possible and impossible observation sets have corresponding orbit states that lie on the critical hypersurface. Proceeding along the solution loop in λ-$\vec{x}$ space, as the corresponding orbit state passes through a point on the critical hypersurface, the observation space image of the orbit state turns back onto another "layer," retracing previous observation sets as λ varies. Thus, at the turning points of the loop in λ-$\vec{x}$ space, that is, the points where

$$\frac{d\lambda}{ds} = 0$$

the $\vec{x}$-component of this point is on the critical hypersurface. These turning points are called critical points. The equivalent condition, that the determinant of the observations partials matrix vanish at the turning points, can also

be obtained by considering the rate of change of the left-hand sides of Equation (3-3) along a solution loop:

$$\frac{d}{ds}\left[o_i^0 + \lambda\left(o_i^1 - o_i^0\right) - C_i(\vec{x})\right] = 0, \quad i = 1, \ldots, 6$$

or

$$\left(o_i^1 - o_i^0\right)\frac{d\lambda}{ds} - \sum_{j=1}^{6} \frac{\partial C_i}{\partial x_j}\frac{dx_j}{ds} = 0, \quad i = 1, \ldots, 6 \qquad (6\text{-}6)$$

At a point such that $d\lambda/ds = 0$, the matrix $\partial C_i/\partial x_j$ must be singular.

6.2.2 CONNECTION OF DISJOINT SOLUTION LOOPS USING CRITICAL SURFACE PATHS

The idea that paths of orbit states on the critical surface can connect a critical point on one solution loop with a critical point on another disjoint solution loop arises from the property of continuity. In a sequence of orbit determination problems in which (only) the a priori estimate is changed continuously, so as to cause a loop to pinch off as in Figure 3.2, it is reasonable to expect that the two corresponding critical points, forming at the pinch-off point, smoothly move apart as the a priori estimate smoothly changes. Abrupt jumps in the shapes of the disjoint loops are not expected. Under this assumption, there will be a path of critical points that leads from one critical point on one loop to the critical point on the other. Pinched-off loops and a connecting path are schematically illustrated by Figure 6-3.

The connection on the critical hypersurface was numerically tested in a number of cases that had (mildly) disjoint

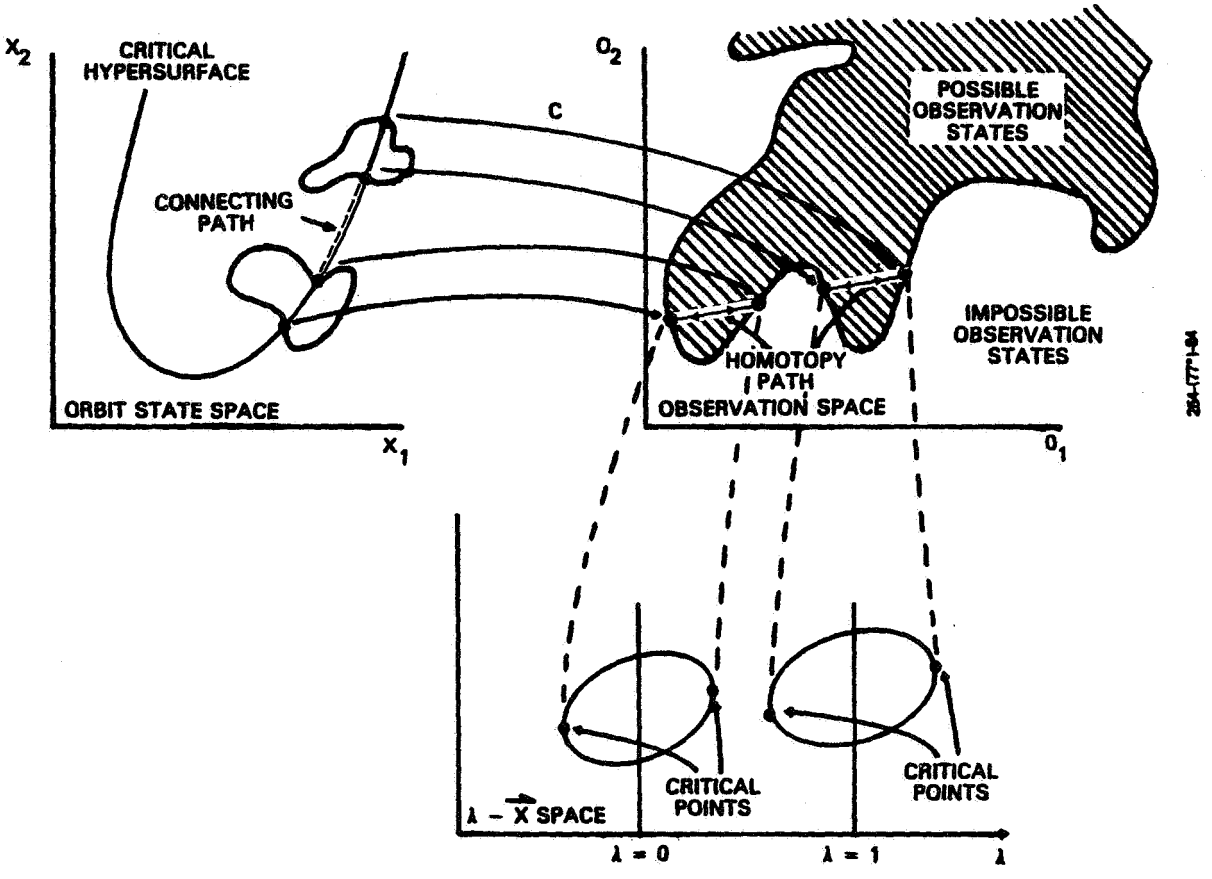


Figure 6-3. Schematic Diagram Showing Pinched-Off Solution Loops and a Connecting Path on the Critical Hypersurface

solution loops. An interactive computer program was written so that paths of small steps on the critical hypersurface could be constructed. Given five coordinates on the hypersurface (any five of x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$) the program solved for the sixth, using the Newton-Raphson iteration on Equation (6-5). The interactive user could, with this program, try to construct a sequence of small steps leading from the known critical point on one solution loop to a known critical point on another loop.

Using this program, it was empirically verified that such continuous paths could indeed be constructed, though the paths sometimes had to be deviously curved. This result, that the critical hypersurface is connected, is the basis for the algorithms of Sections 6.2.3 and 6.2.4.

6.2.3 TWO-LEVEL ALGORITHM

To get around the indentation in the region of possible observation sets in Figure 6-3, the homotopy path must deviate from the straight line path that is indicated there. In the two-level algorithm, the connecting path is allowed to move freely in a plane in observation space. The observation sets in this plane are labeled by two parameters, λ_1 and λ_2. The additional degree of freedom is cancelled by the constraint that the orbit state must lie on the critical hypersurface, so that the equations still define a curve. Thus, the two-level algorithm is based on the following two sets of equations:

Level 1 Loops

$$o_i^0 + \lambda_1\left(o_i^1 - o_i^0\right) - C_i(x) = 0, \quad i = 1, \ldots, 6 \qquad (6\text{-}7)$$

$$\lambda_2 = 0$$

Level 2 Loops

$$O_i^0 + \lambda_1\left(O_i^1 - O_i^0\right) + \lambda_2\left(O_i^2 - O_i^0\right) - C_i(\vec{x}) = 0,\ i = 1, \ldots, 6$$

$$\det\left[\frac{\partial C_k}{\partial x_j}\right]_{\vec{x}} = 0 \qquad (6\text{-}8)$$

Level 1 loops are exactly those defined by Equations (3-3) except that λ has been renamed λ_1. Level 2 loops are defined in an eight-dimensional $(\lambda_1\text{-}\lambda_2\text{-}\vec{x})$-space and the determinant condition forces the orbit states of level 2 loops to lie on the critical hypersurface. Level 1 loops can be considered to lie in the $\lambda_2 = 0$ hypersurface of the eight-dimensional $(\lambda_1\text{-}\lambda_2\text{-}\vec{x})$-space. The term $O_i^2 - O_i^0$ is the i-th component of a vector in observation space that is chosen to be linearly independent of the vector with components $O_i^1 - O_i^0$, $i = 1, \ldots, 6$. The two six-dimensional vectors $(\vec{O}^2 - \vec{O}^0)$ and $(\vec{O}^1 - \vec{O}^0)$ define the plane of the homotopy path. The observations O_i^2 may be arbitrarily chosen; one method is to choose a second a priori estimate, $\vec{x}^2$, and then set

$$O_i^2 = C_i(\vec{x}^2),\ i = 1, \ldots, 6 \qquad (6\text{-}9)$$

The computational procedure for the two-level algorithm is as follows:

Step 1: Compute the first level 1 loop, beginning with the a priori estimate $\vec{x}^0$ and using Equations (6-7). Save all critical points found on the loop.

Step 2: Beginning with each critical point from Step 1, and beginning at $\lambda_2 = 0$, calculate all corresponding, distinct level 2 loops. Save all critical points on these loops (defined by $\lambda_2 = 0$). The level 2 loops are calculated from Equations (6-8).

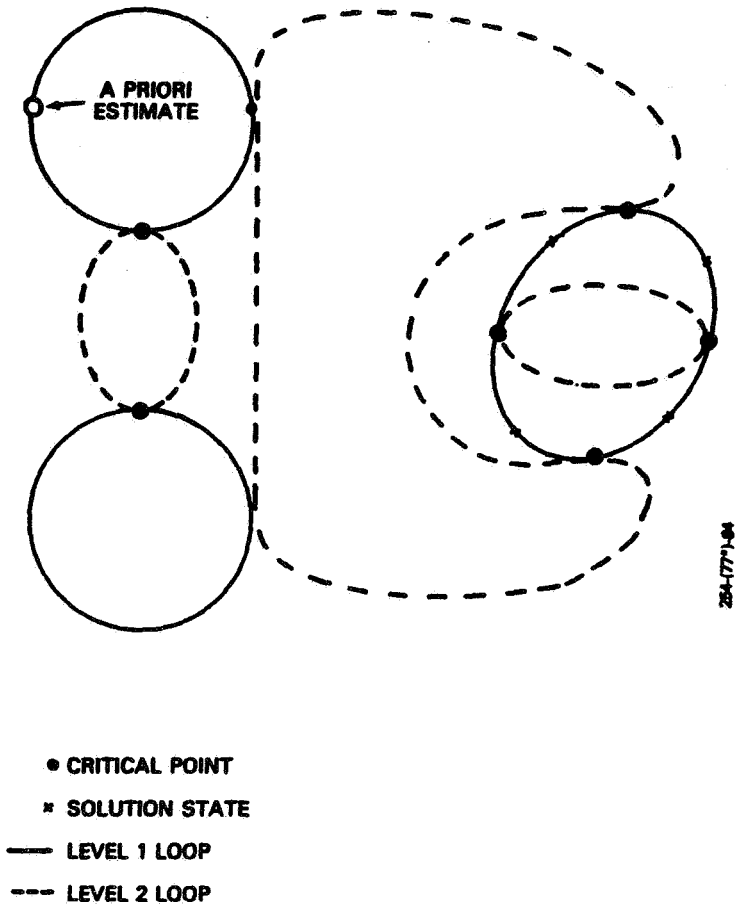
Step 3: If new critical points are found on the level 2 loops of Step 2, then calculate the corresponding, distinct level 1 loops beginning at these critical points.

Step 4: Iteratively repeat Steps 2 and 3 until each critical point on a level 1 loop is matched with a corresponding critical point on a level 2 loop.

Of course, all solution states at $\lambda = 1$ are collected during calculation of the level 1 loops.

The computational procedure is automatically guided by construction of a critical-point table. Each critical point for each loop is added to the table as each additional level 1 or level 2 loop is completed. Pointers in the table indicate whether a given critical point on a level 1 loop is matched by a corresponding critical point on a level 2 loop or vice versa, or whether a given critical point is unmatched. If the pointers indicate an unmatched critical point, the appropriate loop calculation is initiated. The table is checked and updated after each loop calculation until all critical points are matched. The network of level 1 and level 2 loops is then complete.

Two examples, A and B, are schematically illustrated by Figures 6-4 and 6-5. The schedule of simulated tracking (single TDRS), the a priori estimates, and the measurement errors for these two examples are listed in Table 6-1. Example A has three level 1 loops and three level 2 loops,



• CRITICAL POINT

× SOLUTION STATE

—— LEVEL 1 LOOP

--- LEVEL 2 LOOP

Figure 6-4. Schematic Diagram of the Solution Curve Network for Example A (Two-Level Algorithm)

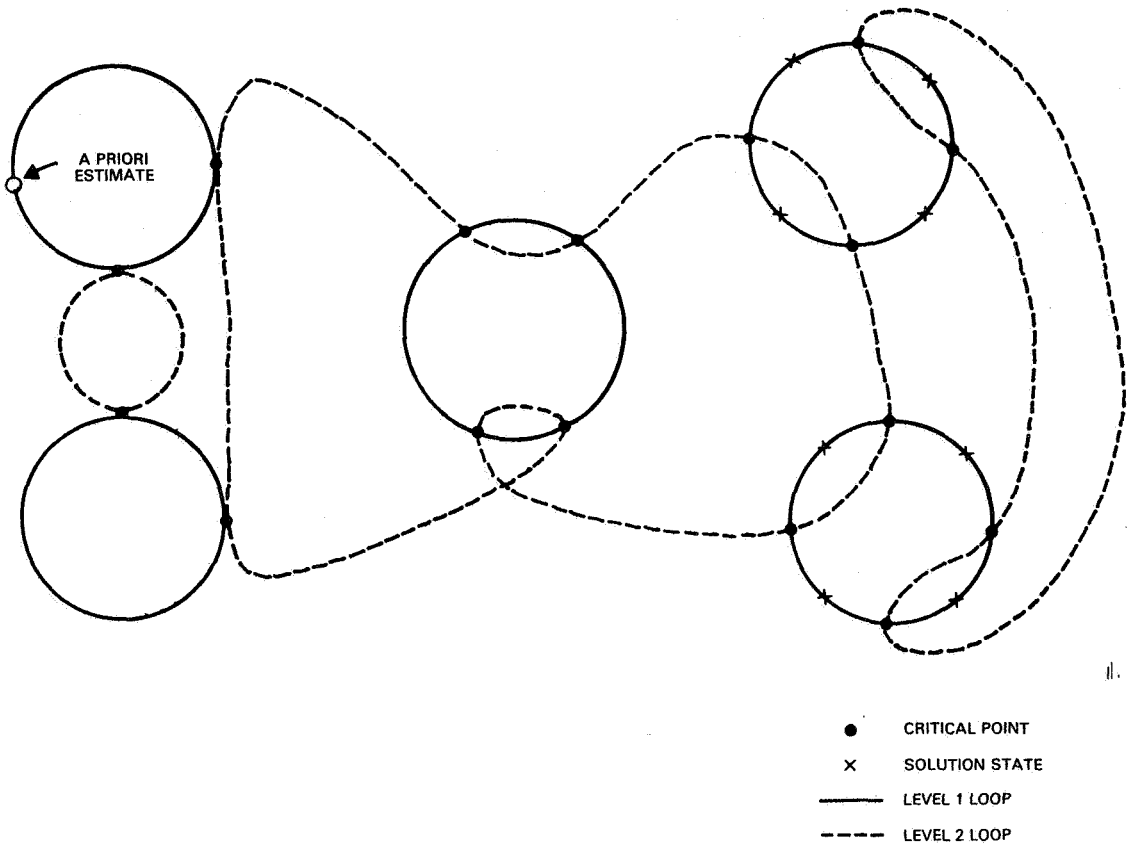


Figure 6-5. Schematic Diagram of the Solution Curve Network for Example B (Two-Level Algorithm)

Table 6-1. Truth and A Priori Estimates, Tracking Schedule, and Measurement Errors for Examples A and B

	Example A			Example B		
Kepler Element	Truth State	A Priori Estimate 1	A Priori Estimate 2	Truth State	A Priori Estimate 1	A Priori Estimate 2
a (km)	7000	8000	8000	7000	10,000	10,000
e	0.001	0.02	0.02	0.001	0.02	0.02
i (deg)	100	65	65	100	65	65
Ω (deg)	55	0	0	55	0	0
ω (deg)	0	0	0	0	0	0
M (deg)	-10	0	30	-100	0	30

Examples A and B

Tracking Schedule: Three Range-Doppler pairs at T = 0, 25, 95 minutes

Measurement Errors: σ(Range) = 5×10^{-4} kilometers
σ(Doppler) = 5×10^{-7} kilometers per second
Bias (Range) = 0.01 kilometers

connected through eight critical points. There are four solution states on one of the level 1 loops. Example B has five level 1 loops and three level 2 loops, connected through 16 critical points. Eight solution states are found on two level 1 loops. In each of these examples, the a priori state and the solution states lie on disjoint level 1 loops. Example A is typical of the complexity of the two-level networks encountered during testing; Example B is more complex than usual.

6.2.4 SIX-LEVEL ALGORITHM

The two-level algorithm can fail; that is, for some a priori estimates, the desired level 1 loop was not found in the computed network. For the method to be fully generalized, the homotopy path must be allowed to lie in a six-dimensional space, rather than only in some two-dimensional subspace. The six-level algorithm is constructed to accomplish this. Although the six-level algorithm was not tested in this study, it is briefly formulated here to indicate a starting point for future work.

The six-level algorithm includes the calculation of solution loops in $(\lambda_1\text{-}\lambda_2\ldots\text{-}\lambda_6\text{-})\vec{x}$-space and in various lower-dimensional spaces. The hierarchy of loops and critical points is listed in Table 6-2. The loops of neighboring levels are connected at critical points of appropriate type. The critical points discussed for the two-level algorithm are, specifically, the level 1 critical points in Table 6-2. They are identified on level 1 loops as points that satisfy either

$$\det\left[\frac{\partial c_k}{\partial x_j}\right] = 0 \qquad (6\text{-}10)$$

Table 6-2. Hierarchy of Loops in the Six-Level Algorithm

Loop Type	Rank of Observation Partials Matrix on Most of Loop	Critical Point Types and Corresponding Rank of Observation Partials Matrix at Critical Point	Dependent Variables on Loop	Constrained Variables on Loop
Level 1	6	Level 1, Rank = 5	λ_1, x	$\lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$
Level 2	5	Level 1, Rank = 5 Level 2, Rank = 4	λ_1, λ_2, x	$\lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$
Level 3	4	Level 2, Rank = 4 Level 3, Rank = 3	$\lambda_1, \lambda_2, \lambda_3, x$	$\lambda_4 = \lambda_5 = \lambda_6 = 0$
Level 4	3	Level 3, Rank = 3 Level 4, Rank = 2	$\lambda_1, \lambda_2, \lambda_3, \lambda_4, x$	$\lambda_5 = \lambda_6 = 0$
Level 5	2	Level 4, Rank = 2 Level 5, Rank = 1	$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, x$	$\lambda_6 = 0$
Level 6	1	Level 5, Rank = 1 (Rank = 0 Points Not Collected)	$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \lambda_5, \lambda_6, x$	None

or, equivalently,

$$\frac{d\lambda_1}{ds} = 0 \tag{6-11}$$

and they are identified on level 2 loops by

$$\lambda_2 = 0 \tag{6-12}$$

More generally, level n critical points are identified on level n loops as the points that satisfy either

$$S_n = 0 \tag{6-13}$$

or, equivalently,

$$\frac{d\lambda_1}{ds} = \frac{d\lambda_2}{ds} = \ldots = \frac{d\lambda_n}{ds} = 0 \tag{6-14}$$

and they are identified on level n+1 loops by

$$\lambda_{n+1} = 0 \tag{6-15}$$

The symbol S_n denotes the sum of all of the n-rowed principal minors of the observations partials matrix. Thus, for example, S_1 is the trace of that matrix and S_6 is the determinant. The S_n's are the coefficients in the characteristic polynomial, $P(\eta)$, of the observations partials matrix (see Reference 11).

$$P(\eta) = \eta^6 + \sum_{k=1}^{6} (-1)^k S_k \eta^{6-k} \tag{6-16}$$

and the characteristic values η_n, n = 1, ..., 6 are the six roots of

$$P(\eta) = 0 \tag{6-17}$$

The observations partials matrix has rank 6-m ($0 \leq m \leq 6$) if and only if m of the eigenvalues are zero, and this happens if and only if $S_6 = S_5 = \ldots = S_{6-m+1} = 0$, with $S_{6-m} \neq 0$.

Finally, the equations for the loops of each level are given. They are

Level 1:

$$\begin{aligned} O_i^0 + \lambda_1\left(O_i^1 - O_i^0\right) - C_i(\vec{x}) &= 0 \qquad i = 1, \ldots, 6 \\ \lambda_i &= 0 \qquad i = 2, \ldots, 6 \end{aligned} \tag{6-18}$$

Level n ($1 < n \leq 6$):

$$\begin{aligned} O_i^0 + \sum_{k=1}^{n} \lambda_k\left(O_i^k - O_i^0\right) - C_i(x) &= 0, \; i = 1, \ldots, 6 \\ S_{8-j} &= 0, \; j = 2, \ldots, n \\ \lambda_i &= 0, \; i = n+1, n+2, \ldots, 6 \end{aligned} \tag{6-19}$$

The middle equation guarantees that (most of) the orbit states on a loop of level n are such that the observation partials matrix has rank 7-n, in accordance with the hierarchy listed in Table 6-2. In Equation (6-19), care must be taken that the vectors $\vec{O}^k - \vec{O}^0$, k = 1, ..., 6 span a six-dimensional space. The computational procedure for the six-level algorithm would be a direct extension of the

procedure for the two-level algorithm. The critical point table, however, must be enhanced to include five types of critical points, and, correspondingly, five types of pointers to indicate critical point matching.

6.2.5 ASSESSMENT

Although the two-level algorithm verifies the technique and the six-level algorithm is expected to provide an exhaustive, systematic algorithm for finding all of the solutions, regardless of the a priori estimate, the technique requires some further, though routine, work before consideration for operational implementation. The reason is efficiency. Each critical surface loop currently requires about 10 times as much CPU time as does an ordinary loop. This is because of the need for partial derivatives of the observation partials determinants in Equation (6-19). These are needed in the corrector part of the curve-following algorithm. In this pilot study, the determinant was simply evaluated at seven points, and its partial derivatives were obtained from numerical differences. This is easy but inefficient, since it should be much faster to use closed-form two-body second-order partial derivatives and then to explicitly compute the partial derivatives of the determinant. Also, the numerical differencing method probably slowed corrector convergence (leading to small steps) through nonoptimum choice of the differencing intervals.

The efficiency problem should be addressed through construction of a subroutine for analytic calculation of the two-body second-order partial derivatives. It is expected that the necessary formulation already exists in the literature; only a careful implementation of it should be required.

SECTION 7 - CONCLUSION

7.1 SUMMARY

TDRSS early orbit determination requires finding orbit solutions with range and/or Doppler data alone, since TDRS antenna pointing angles, which are simply open-loop pointing angles, are not sufficiently accurate. Given six range and/or Doppler observations, the homotopy continuation method, as formulated here, derives the solution by mathematically defining a smooth, continuous curve that extends from the a priori estimate to the solution in a specially defined seven-dimensional space. This solution curve will generally pass through a number of other solutions before returning to the a priori estimate. A numerical algorithm, described in detail in Section 3.5, was developed that allows a computer program to systematically follow such a solution curve completely around the loop, collecting the orbit solutions along the way.

The accuracy of this method is limited solely by the accuracy of the tracking data and physical models selected. Two-body dynamics and geometric range and Doppler modeling yielded one-revolution accuracies of about 50 kilometers for Landsat-4. Use of a Brouwer-Lyddane propagator and observation light time modeling was able to refine these solutions to an accuracy of about 2 kilometers. Thus, the homotopy continuation method has the flexibility to permit the use of physical models of increasing accuracy.

CPU time requirements for the method are fast enough to permit operational use. In the developmental program on the VAX 11/780 computer, a typical solution loop required 0.5 to 2 minutes for its computation. The accuracy refinement stage, if invoked, requires about the same time. An operational version of this program would contain improvements

for significantly faster operation. Thus, the method is fast enough for online use in orbit operations.

A prototype automatic early orbit algorithm was devised and its critical features were tested. This algorithm develops preliminary orbit estimates from the TDRS antenna pointing angles, followed by orbit solutions derived from the precise range and Doppler tracking. In testing, this algorithm obtained the correct solution in all but 4 of 420 Monte Carlo trials. The four failures resulted from premature termination of the algorithm after finding one or more extraneous solutions that also satisfied the acceptance criteria specified. This deficiency can be corrected through the use of more stringent acceptance criteria, for example, additional checking of tracking observations, antenna pointing angles and grass orbital parameters.

This study has not considered in detail the question of how much TDRSS range/Doppler tracking is necessary to determine an orbit with sufficient accuracy. This question is essentially independent of the method used to find an orbit solution for specified tracking, and can be handled by standard error analysis techniques. However, the results obtained for Landsat-4 do indicate that, for a low-altitude satellite, 15 minutes is too little tracking (using only one TDRS), while one orbital period is sufficient. Unfortunately, intermediate amounts of Landsat-4 TDRSS tracking were not available, so that further examination of this question must be left for future work.

7.2 SUGGESTIONS FOR FURTHER DEVELOPMENT

7.2.1 EXTENDED HOMOTOPY CONTINUATION METHOD

The only significant problem inherent in the basic homotopy method, as formulated in Section 2, is that, if the a priori state is very different from the solution state, the two

states may lie on disjoint solution loops, making it impossible to reach the solution in a straightforward way. In the prototype early orbit algorithm, this problem was avoided by deriving sufficiently good a priori estimates from the antenna pointing angles. Nevertheless, in other applications, it may be useful to have an algorithm that will work from any a priori estimate. An algorithm that is believed to have this property was devised and described in Section 6.2. By suitable generalization, this extended algorithm defines and computes a connected network of loops, enabling the numerical algorithm to reach solution loops that are disjoint in the basic formulation.

Further development should include testing of the full six-level algorithm (only a limited two-level version was testing in this study) to determine if it is truly global and exhaustive as expected. Development of a global and exhaustive algorithm for the range/Doppler problem can be significant for orbit determination in general because it could eliminate the need to collect antenna pointing measurements, shifting the burden to the computer, and thus simplifying the overall system. Furthermore, such an algorithm could perform early orbit determination in systems with very broad antenna beam patterns.

7.2.2 LEAST-SQUARES ORBIT DETERMINATION

Use of the least-squares method for early orbit determination has two principal advantages over a six-observation method. First, a least-squares method will reduce the sensitivity of the solution to high-frequency noise in the tracking data. Second, the addition of a Bayesian term in the formulation permits a priori knowledge of orbit parameters to be included in the solution, which allows valid solutions with smaller amounts of tracking.

The negative feature of a least-squares method is that more extraneous solutions may be introduced because of the mathematical nature of the least-squares formulation.

In a homotopy formulation, least-squares orbit determination leads to solution curves that are more complex than the simple loops of the six-observation problem. These complex curves contain bifurcation points at which several curve components are joined together. The numerical algorithm used in this work for following solution curves stalls at these bifurcation points, and, thus, the least-squares method could not be systematically studied at the present time.

For future work, three methods for handling the bifurcation points might be studied. One method would use an analytical patch for describing the solution curve near such a point and would use the current numerical algorithm on the regular part of the curve. Another approach is to abandon the current numerical algorithm and use a simplicial method to follow the curve (Reference 4). A simplicial method constructs a "triangulation" network to enclose the curve and to proceed along it. Some simplicial methods may be capable of proceeding through the bifurcation points without difficulty. Furthermore, a simplicial method may be useful in addressing the efficiency problem in the enhanced (six-level) continuation method, since the higher-order derivatives would not be required. Finally, the third approach for study is the method of George (Reference 8) for locally perturbing the problem so that the bifurcation temporarily disappears. Development of the capability for an <u>autonomous</u> computer program to systematically handle the bifurcation points on a solution curve would provide a major, significant step in the development of the homotopy method, considerably enlarging its domain of applicability.

REFERENCES

1. Escobal, P. R., Methods of Orbit Determination, New York: John Wiley and Sons, Inc., 1965

2. Chow, S. N., J. Mallet-Paret, and J. A. Yorke, "Finding Zeros of Maps: Homotopy Methods That Are Constructive With Probability One," Mathematics of Computation, vol. 32, pg. 887, 1978

3. Watson, L. T., "Engineering Applications of the Chow-Yorke Algorithm," Applied Mathematics and Computation, vol. 9, pg. 11, 1981

4. Allgower, E., and K. George, "Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations," SIAM Review, vol. 22, pg. 28, 1980

5. Goddard Space Flight Center, TDO 78-43, TDRS Observed and Computed Measurement Algorithms, J. Teles and V.S. Guedney, April 1978

6. Goddard Space Flight Center, STDN No. 721, Tracking Data Transmission Formats and Reduction Algorithms, October 1980

7. Goddard Space Flight Center, X-582-76-77, Mathematical Theory of the Goddard Trajectory Determination System (GTDS), A. J. Fuchs and C. E. Velez, with J. O. Cappellari, Jr., of Computer Sciences Corporation (eds.), April 1976

8. Smith, R. L., and C. Y. Huang, "A Homotopy Continuation Method for General Preliminary Orbit Determination and Special Application to the Tracking and Data Relay Satellite System", Paper No. 85-0215, Presented at the AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 14-17, 1985

9. Georg, K., "On Tracing an Implicitly Defined Curve by Quasi-Newton Steps and Calculating Bifurcation By Local Perturbation", SIAM Journal on Scientific and Statistical Computing, Vol. 2, no. 1, pp. 35-50, 1981

10. Abramowitz, Milton, and Irene A. Stegun, Handbook of Mathematical Functions, Equation 25.4.29, New York: Dover Publications, Inc., 1965

11. Mirsky, L., An Introduction to Linear Algebra, pg. 198, New York: Dover Publications, Inc., 1982

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-86230		

4. Title and Subtitle	5. Report Date
STUDY OF A HOMOTOPY CONTINUATION METHOD FOR EARLY ORBIT DETERMINATION WITH THE TRACKING AND DATA RELAY SATELLITE SYSTEM (TDRSS)	March 1986
	6. Performing Organization Code: 554
7. Author(s): R. L. Smith and C. Huang	8. Performing Organization Report No.
9. Performing Organization Name and Address: Goddard Space Flight Center, Greenbelt, Maryland 20771	10. Work Unit No.: 85BO551
	11. Contract or Grant No.
	13. Type of Report and Period Covered: Technical Memorandum
12. Sponsoring Agency Name and Address: NASA, Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes: Computer Sciences Corporation, Beltsville, MD. R. L. Smith and C. Huang

16. Abstract

A recent mathematical technique for solving systems of equations is applied in a very general way to the orbit determination problem. The study of this technique, the homotopy continuation method, was motivated by the possible need to perform early orbit determination with the Tracking and Data Relay Satellite System (TDRSS), using range and Doppler tracking alone. Basically, a set of six tracking observations is continuously transformed from a set with known solution to the given set of observations with unknown solutions, and the corresponding orbit state vector is followed from the a priori estimate to the solutions. A numerical algorithm for following the state vector is developed and described in detail. Numerical examples using both real and simulated TDRSS tracking are given. A prototype early orbit determination algorithm for possible use in TDRSS orbit operations was extensively tested, and the results are described. Preliminary studies of two extensions of the method are discussed: generalization to a least-squares formulation and generalization to an exhaustive global method.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Homotopy, Continuation Method, TDRSS, Early Orbit Determination, Non-linear Equations Method.	Unclassified Unlimited Subject Category 13

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	140	A07

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986